Naval Research Laboratory

Washington, DC 20375-5320

NRL/FR/7140--95-9723

BiRASP - The Bistatic Range-dependent Active System Performance Prediction Model

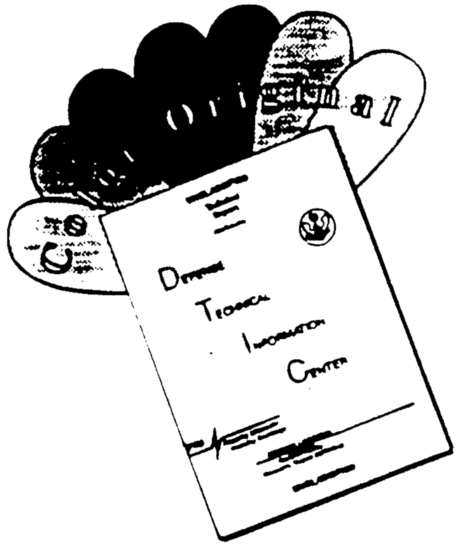
DAVID MELOY FROMM
JOHN P. CROCKETT
L. BRUCE PALMER

Acoustics Systems Branch
Acoustics Division

September 30, 1996

Approved for public release; distribution unlimited.

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

DEPARTMENT OF THE NAVY
NAVAL RESEARCH LABORATORY
4555 OVERLOOK AVE SW
WASHINGTON DC 20375-5320

IN REPLY REFER TO

1022157

5231-091A:PS
22 Oct 1996

From: Commanding Officer, Naval Research Laboratory

To: Distribution

Subj: ERRATA II; NRL/FR/7140--95-9723, "BiRASP - The Bistatic Range-dependent Active System Performance Prediction Model," by David M. Fromm, et al., dated September 30, 1996.

1. In NRL/FR/7140--95-9723, the title on the cover is incorrect. The correct title is "BiRASP - The Bistatic Range-dependent Active System Performance Prediction Model."

2. Please make a pen-and-ink correction to the title on the cover.

3. The correct title is on the Report Documentation Page, page i, and on the first page of the report itself, (page 1).

4. We regret any inconvenience.

Timothy D. Calderwood
Publication Officer

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 30, 1996	

4. TITLE AND SUBTITLE

BiRASP — The Bistatic Range-dependent Active System Performance Prediction Model

5. FUNDING NUMBERS

PE - 62435N
UW 035-203

6. AUTHOR(S)

David Meloy Fromm, John P. Crockett, and L. Bruce Palmer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/FR/7140--95-9723

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
Arlington, VA 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In 1992, the Naval Research Laboratory (NRL) published a report describing the Range-dependent Active System Performance (RASP) prediction model; a sequence of computer programs using multipath propagation and scattering processes to predict the long-range, low-frequency boundary reverberation, and target returns that would be received in real ocean environments. That computer model has been extended to admit arbitrary source/receiver configurations within a three-dimensional, range-dependent environment. The enhancements include: (1) range and azimuthal dependence in all environmental parameters, (2) volume scattering and bistatic scattering strength functions, (3) realistic source and receiver characteristics (e.g., three-dimensional beam patterns for linear arrays and array tilt), and (4) calculation of target returns with time/angle spreading. In addition to addressing the prediction and evaluation of active sonar systems in real ocean environments, the set of programs that comprise the Bistatic Range-dependent Active System Performance (BiRASP) prediction model also produce the detailed, high-resolution results required for basic theoretic and experimental acoustic research. This report presents the theoretical foundations of the BiRASP model and the corresponding numerical implementation of this theory. Further, a detailed description of the model software, instructions for execution, and sample results are provided.

14. SUBJECT TERMS

Active surveillance	Ocean boundary scattering	Three-dimensional beam patterns
Bistatic boundary scattering	Propagation	Three-dimensional ocean environments
Bistatic geometry	Ray acoustics	Transmission loss
Monostatic geometry	Reverberation vs beam vs time	Volume scattering

15. NUMBER OF PAGES

97

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18
298-102

CONTENTS

LIST OF FIGURES

LIST OF TABLES

BiRASP – THE BISTATIC RANGE-DEPENDENT ACTIVE SYSTEM PERFORMANCE PREDICTION MODEL

1. INTRODUCTION

The performance of an active sonar is profoundly affected by the ocean environment in which it operates. When a low-to-mid frequency acoustic pulse is transmitted in the deep ocean, it tends to propagate over nearly cyclic refractive paths to long ranges while portions of the signal energy interact with the ocean boundaries, ocean-volume scatterers, and, hopefully, a target of interest. The boundary interactions and volume scatterers give rise to reverberant returns that typically exceed ambient noise and may mask target echoes at a receiver. Thus, to accurately predict the active system performance in real ocean environments, it is necessary to comprehensively account not only for system parameters but also for the complex processes of acoustic propagation and scattering. This report describes a computer model that accomplishes this. The model is designed for, but not necessarily limited to, deep ocean environments.

1.1 Background

In 1992, the Naval Research Laboratory (NRL) published a report [1] describing a comprehensive computer model designed to predict target echos and boundary reverberation that would be received in real ocean environments. The model, which encompasses realistic multipath propagation and scattering processes, is called the Range-dependent Active System Performance (RASP) prediction model. Features of the RASP model include:

- A variety of source and receiver geometries,
  - monostatic (collocation of source and receiver) in range-dependent environments
  - quasi-monostatic (source and receiver separated only in depth) in range-dependent environments
  - bistatic (range- and depth- separated source and receiver) in range-independent environments
- All major multipath contributors to reverberation with the application of a wave-theoretic caustic treatment at smooth caustics,
- Source and receiver vertical beam patterns,
- Finite pulse lengths and system processing parameters such as bandwidth and analysis time,
- The prediction of target returns as a function of range for up to three target depths,
- Commonly used models of angle-dependent boundary backscattering strengths (Chapman-Harris [2], and the Urick-Mackenzie [3,4]) and bottom loss (Fleet Numeric Weather Central (FNWC) bottom types [5]),

Manuscript approved May 24, 1996.

- Allowances for angle- and frequency-dependent bottom-backscattering strength and bottom-loss functions that vary with range, and
- Implementation in a modular structure that facilitates upgrades and extensions.

This report documents the bistatic extension of the RASP model. The capabilities of the Bistatic Range-dependent Active System Performance (BiRASP) prediction model have been extended to include:

- Arbitrary monostatic and bistatic source/receiver geometries in range-dependent environments. Multistatic geometries can also be constructed.
- Realistic source/receiver array characteristics: three-dimensional (3D) beam patterns for linear arrays, spatial shading and dead elements, array tilt.
- The Ogden-Erskine surface backscattering strength model [6] and the Ellis bistatic bottom scattering model [7].
- Calculation of volume reverberation.
- Arbitrary 3D range-dependent environments, specified by N bearing-dependent (radial) two-dimensional (2D) range-dependent environments ($N \times 2\text{D}$). The environmental models are not tied to specific databases; rather, user-supplied auxiliary programs must be used to access standard databases for sound speed, bathymetry, and bottom loss. [8–10].
- An optimized integration algorithm with error control to minimize execution time.
- Calculation of transmission loss, target echo, and signal excess with or without time-angle spreading.
- Calculation of launch/arrival/grazing angle distributions of transmission loss, reverberation, and target echo.

Much of the material of Ref. 1 has been either revised, summarized, or duplicated in this report. This is a result of the BiRASP model being an evolution of the RASP model and an attempt to allow the reader to exercise the BiRASP model based solely on the information contained in this report.

1.2 Purpose

The purpose of this report is to document the BiRASP model, the procedures for executing the model, and its application to analysis of low-to-mid frequency active sonar concepts. At NRL, the model resides on a Silicon Graphics Iris (SGI) color workstation. The graphics are a combination of the DISSPLA®plotting package by Computer Associates and in-house color graphics software on the SGI. The theoretical formulations implemented in the BiRASP model are presented in the next section, and general model overview is given in the third section. In the fourth section, each of the individual program modules are described in detail. The last section presents a sample BiRASP execution. A detailed listing of the input variables for each program module is provided as an Appendix.

2. THEORETICAL FORMULATIONS

In this section, we formulate expressions for boundary and volume reverberation and for target echo. The basic approach was first reported in Ref. 11 and extended in Ref. 1. The following reflects modifications for the current model, which has been extended to include volume reverberation. The treatment is for arbitrary bistatic source/receiver geometries. Monostatic and quasi-monostatic source/receiver geometries are considered as special cases.

2.1 Boundary Reverberation

Assume that a point source S radiates a time-dependent acoustic signal in a 3D ocean with the acoustic energy propagating away from S along ray paths. As a ray encounters an ocean boundary, it continues to propagate in the direction of specular reflection minus a small amount of radiation dispersed according to a scattering law. The scattering can be viewed as arising from excitation of small elements in the boundary, each of which acts as a weak source. The sum of the scattered radiation detected at the receiving point R is the boundary reverberation level at any given time.

To calculate boundary reverberation, it is first necessary to trace a set of acoustic rays from S and from R; recording travel times, intensities, and other parameters at each boundary encounter. To proceed, four specific assumptions regarding scattering are made:

1. Scattering surfaces can be decomposed into elemental surfaces that are secondary point-sources while they are ensonified. Intensities of secondary sources are proportional to the incident intensity, the elemental area dA, and the scattering strength.
2. In determining a mean reverberation envelope, any interference effects associated with acoustic phase difference can be ignored. Scattered rays would therefore be represented as having, in effect, a random phase shift relative to the incident ray. Such an approach results in a mean reverberation envelope representative of ensemble-averaged returns. In principle, a coherent summation would lead to an envelope more indicative of a single sample return, but this would require a time-dependent representation of the surface.
3. The scattering layer at the ocean surface can be approximated by a horizontal plane with an appropriate scattering coefficient, and similarly the bottom scattering surface is approximated by the (gross) bottom topography with an appropriate scattering coefficient.
4. The direction of propagation and of the reflected energy is confined to the vertical plane containing the incident ray path. This last assumption removes the requirement for a fully 3D acoustic propagation model.

For each element of the ocean boundary from which scattering reverberation is to be calculated, there is an associated set of reverberant paths, each making an elemental contribution to the reverberation. The contributions are each composed of the transmission losses associated with the ray path from the source to the scattering element and back to the receiver, being appropriately weighted by the source and receiver beam patterns, the scattering strength, and the area of the element. The expected (or averaged) value of the instantaneous reverberation is then the sum of all of the elemental contributions active at that instant.

A bistatic geometry is that in which the source and receiver are separated in range and possibly separated in depth. Figure 1 depicts surface reverberation with a source S that ensonifies a scattering element dA located at (x, y) via a set of paths indexed by $m(x, y)$ and a receiver R that receives a return via a different set of paths indexed by $n(x, y)$. First, we consider the contribution to the reverberation due to a single pair of paths; $i \in m(x, y)$ and $j \in n(x, y)$. Presuming S emits a time-varying signal of an intensity $I(t)$, weighted by a beam pattern B, the intensity is reduced by the transmission loss L_i along path i from S to dA. The incident intensity at dA is then weighted by the scattering strength σ. The return is further diminished by the transmission loss $\widetilde{L}_j$ along

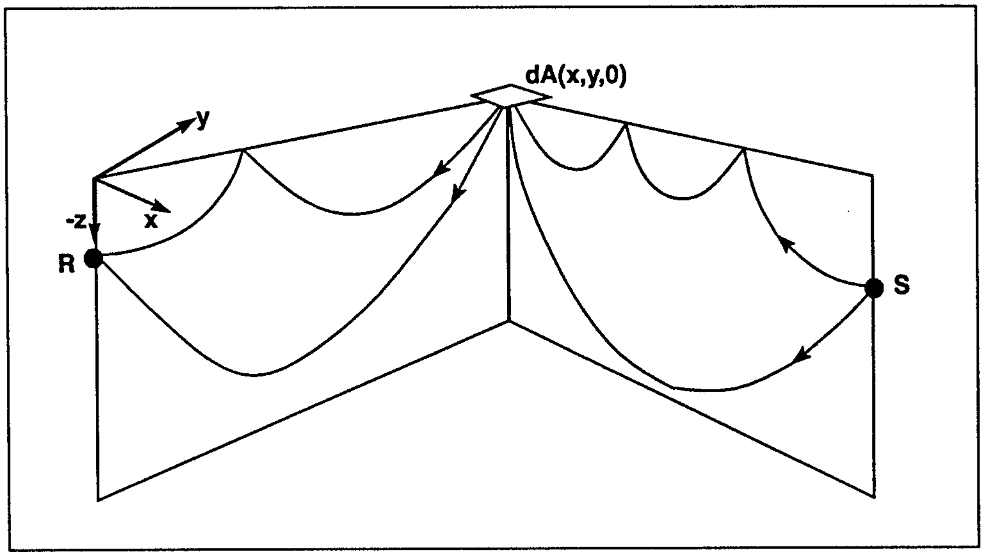


Fig. 1 – Source ensonifying a scattering element dA via propagation paths indexed by $m(x, y)$ and a receiver receiving reverberation via return paths indexed by $n(x, y)$

path j from dA to R. At R, it is weighted by the receiver beam pattern $\widetilde{B}$. The time delay for the reverberation is simply the two-way travel time $T_i + \widetilde{T}_j$. That is, the unweighted initial source intensity that results in reverberation at time t, due only to outward path i and return path j, is $I(t - T_i - \widetilde{T}_j)$. Summing over all multipath combinations from S to dA to R, the total contribution from dA is obtained. These intensities are to be summed with respect to time but without regard to phase due to the second assumption. Finally, by integrating over all scattering elements, the total reverberation $R(t)$ at t is found to be

$$R(t) = \int\int_A \sum_{i \in m(x,y)} \sum_{j \in n(x,y)} I[t - T_i(x,y) - \widetilde{T}_j(x,y)] \, \frac{B(\theta_i, \phi_i) \, \widetilde{B}(\widetilde{\theta}_j, \widetilde{\phi}_j)}{L_i(x,y) \, \widetilde{L}_j(x,y)} \, \sigma(\theta'_i, \phi_i, \widetilde{\theta'}_j, \widetilde{\phi}_j) \, dA \; . \quad (1)$$

The azimuthal angles ϕ and $\widetilde{\phi}$ are measured in the horizontal plane about the locations of the source and receiver, respectively. The source ray's launch angles θ, and the receiver ray's arrival angles $\widetilde{\theta}$ are measured from the horizontal plane at their respective depths. Note that the scattering strength $\sigma(\theta'_i, \phi_i, \widetilde{\theta'}_j, \widetilde{\phi}_j)$ in Eq. (1) depends on the azimuthal angles and the elevation (i.e., θ' and $\widetilde{\theta'}$) angles of incidence and scatter. If necessary, the scattering strength can also be made range-dependent. At this level of generality, a full 3D scattering pattern is implied.

If we take the signal emitted by S to be a wave of constant intensity I, and duration D, that is turned on at S when $t = 0$, then a scattering element at position (x, y) is actively contributing to reverberation at time t via a propagation path i and return path j if

$$0 \leq t - T_i(x,y) - \widetilde{T}_j(x,y) \leq D \; . \quad (2)$$

The double inequality of Eq. (2) is equivalent to

$$t - D \leq T_i(x,y) + \widetilde{T}_j(x,y) \leq t \; . \quad (3)$$

Therefore, the set $G_{ij}(t)$ of reverberators active at t via the path pair (i, j) is defined by

$$G_{ij}(t) = \{(x,y) \mid t - D \leq T_i(x,y) + \widetilde{T}_j(x,y) \leq t\} \; . \quad (4)$$

The reverberation $R(t)$ from the boundary area A can then be written

$$R(t) = I \int\int_A \sum_{i \in m(x,y)} \sum_{j \in n(x,y)} C_{G_{ij}(t)}(x,y) \, \frac{B(\theta_i, \phi_i) \; \widetilde{B}(\widetilde{\theta}_j, \widetilde{\phi}_j)}{L_i(x,y) \; \widetilde{L}_j(x,y)} \, \sigma(\theta'_i, \phi_i, \widetilde{\theta'}_j, \widetilde{\phi}_j) \; dA \; , \tag{5}$$

where the characteristic function $C_{G_{ij}(t)}(x,y)$ is 1 if $(x,y) \in G_{ij}(t)$ and 0 otherwise.

2.2 Volume Reverberation

It is straightforward to extend Eq. (5) to the case of volume reverberation. Assume that when a ray encounters a volume scatterer, it continues to propagate in its initial direction, minus a small amount of radiation dispersed according to some scattering law. The scattering can be viewed as arising from excitation of small scattering elements in the volume, each of which acts as a weak source. Let $\sigma_V(x,y,z)$ be the omni-directional volume scattering strength, then with the addition of an integral over the depth, Eq. (5) becomes

$$R(t) = I \int\int\int_A \sum_{i \in m(x,y,z)} \sum_{j \in n(x,y,z)} C_{G_{ij}(t)}(x,y,z) \, \frac{B(\theta_i, \phi_i) \; \widetilde{B}(\widetilde{\theta}_j, \widetilde{\phi}_j)}{L_i(x,y,z) \; \widetilde{L}_j(x,y,z)} \, \sigma_V(x,y,z) \; dA \, dz \; . \tag{6}$$

In general, for volume reverberation due to fish, σ_V is nonzero for only a few, relatively thin layers. Let Z_k be the depth of the center of the kth layer with thickness Δz_k. Assuming the energy of each ray can be considered constant across a layer, then the corresponding volume reverberation $R_f(t)$ is given by

$$R_f(t) = I \sum_k \int\int_A \sum_{i \in m(x,y,Z_k)} \sum_{j \in n(x,y,Z_k)} C_{G_{ij}(t)}(x,y,Z_k) \, \frac{B(\theta_i, \phi_i) \; \widetilde{B}(\widetilde{\theta}_j, \widetilde{\phi}_j)}{L_i(x,y,Z_k) \; \widetilde{L}_j(x,y,Z_k)} \, \widehat{\sigma}_k(x,y) \; dA \; , \tag{7}$$

where $\widehat{\sigma}_k(x,y) = \sigma_V(x,y,Z_k)\Delta z_k$ is the scattering strength for the kth layer.

2.3 Target Returns

The preceding formulations can be used to develop expressions for the echo return from a point target by assuming that the target scattering area can be described by a delta function. Thus, it follows from Eq. (5) that the echo return $E(x,y,z_e,t)$ from a point target located at position (x,y) and depth z_e is

$$E(x,y,z_e,t) =$$

$$\sum_{i \in m(x,y)} \sum_{j \in n(x,y)} I[t - T_i(x,y,z_e) - \widetilde{T}_j(x,y,z_e)] \, \frac{B(\theta_i, \phi_i) \; \widetilde{B}(\widetilde{\theta}_j, \widetilde{\phi}_j)}{L_i(x,y,z_e) \; \widetilde{L}_j(x,y,z_e)} \, \sigma(\theta'_i, \phi_i, \widetilde{\theta'}_j, \widetilde{\phi}_j) \; . \tag{8}$$

Note that the scattering strength σ now becomes the target strength. In reality, the target strength may be dependent on parameters in addition to those that appear in Eq. (8), such as target aspect, but that is ignored here.

Ignoring the time dependence and assuming $\sigma(\theta'_i, \phi_i, \widetilde{\theta'}_j, \widetilde{\phi}_j)$ is constant, the target return takes the form of

$$E(x, y, z_e, t) = \sigma \sum_{i \in m(x,y)} \left[\frac{B(\theta_i, \phi_i)}{L_i(x, y, z_e)} \right] \sum_{j \in n(x,y)} \left[\frac{\widetilde{B}(\widetilde{\theta}_j, \widetilde{\phi}_j)}{\widetilde{L}_j(x, y, z_e)} \right] . \tag{9}$$

Equation (9) is a sonar-equation representation except that source and receiver directivities have not been factored out of the summation over ray paths.

2.4 Monostatic and Quasi-monostatic Source/Receiver Geometries

In general, because of the 3D beam patterns of the source and receiver and the arbitrary 3D range-dependent environment, Eqs. (5), (7), and (8) do not significantly simplify algorithmically for monostatic and quasi-monostatic source/receiver geometries.

When it is possible to impose azimuthal symmetry of the environment and the beam patterns about the vertical axis containing the source and receiver, the equations do simplify and result in a significant saving in execution time. An example would be the calculation of the reverberation received in the broadside beam of a horizontal line array due to a vertical source line array. For these cases, replace the location (x, y) with the range ρ and the integration $\int \int_A (\cdots) dA$ with $2\pi \int (\cdots) \rho d\rho$. This "single bearing" evaluation is discussed further in Section 4.5.1.

3. MODEL OVERVIEW

BiRASP is based on range-dependent ray theory, which allows for both range-dependent sound speed fields and bathymetry. The ray-theoretic algorithm of BiRASP is essentially that used by the Germinating Ray-Acoustics Simulation System (GRASS) propagation model [12], which was initially developed at Hudson Laboratories [13]. The calculation of reverberation or target echo first requires the determination of the acoustic ray paths that join either the source or the receiver to each scatterer. Since it is not computationally practical to execute a ray trace to each individual scatterer, BiRASP uses an $N \times 2D$ approach. Two polar coordinate systems, one each centered at the source and receiver locations, are imposed on the environment. About each origin, azimuthal sectors are defined such that the environment can be considered to be slowly varying or constant with respect to azimuth within the sector. Each environmental sector is then sampled along a representative radial, and 2D ray traces are performed. For each ray path, the acoustic intensity, travel time, and incident angle at the scatterers are computed and stored in a look-up table. The reverberation contribution or target echo from a scatterer is obtained by retrieving the ray-trace results for the appropriate source and receiver sectors and summing the time-dependent returns corresponding to all possible combinations of outgoing and returning paths. The total time-dependent reverberation is determined by integrating over the entire scattering region using the polar coordinate system centered on the receiver. Figure 2 summarizes many of the processes and phenomena considered by a BiRASP prediction.

The BiRASP model is a computer implementation of the foregoing ray-theoretic formulations of reverberation and target echoes. It is an assemblage of six programs that execute the primary computations and four support programs to perform the final scaling and create the various plots. In general, the sequence of program executions carry calculations forward from the specification of empirical sound-speed and bathymetry data to the display of predicted levels of surface reverberation, bottom reverberation, volume reverberation, and target echoes as functions time. The six primary computational programs are:

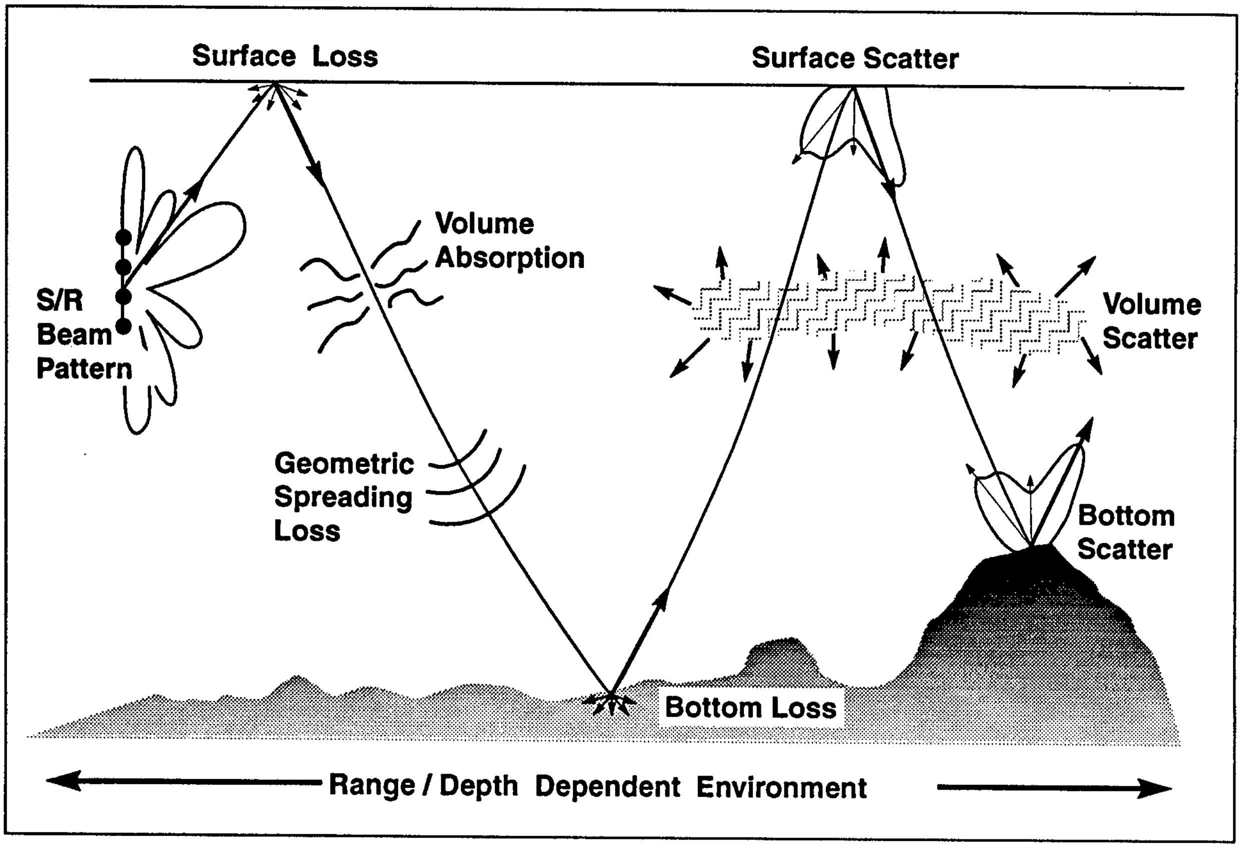


Fig. 2 – Summary of processes and phenomenon considered by a BiRASP prediction

- PROFIL – Constructs the 2D environmental field along a sector radial from empirical data.
- RAYACT – Computes ray paths from the source or receiver to the sea surface, bottom, and up to three combinations of target or volume-layer depths.
- RTHETA – Organizes source- or receiver-originating ray-path encounters with a boundary or target/layer depth (into range-angle order contours).
- TLGRID – Interpolates and maps the acoustic intensity, travel time, and grazing angle computed in RTHETA for each ray path along a radial onto a uniform-spaced radial-range grid. Further data manipulation (e.g., sorting by acoustic intensity) is performed.
- BIREV – Computes raw (uncalibrated) reverberation from a single boundary or volume layer as a function of (some) angle and time. Also, target returns can be calculated either for a single, fixed-target location or as though a target is located at each elemental range-azimuthal scattering patch. The angle parameter of computed envelopes is typically receive or arrival angle but can be one of several possibilities, including launch angle, bistatic-angle, and grazing angle. The versatility in specifying the dependent angle is included as an analysis tool.

- POSTBIREV – Applies beam patterns (for multiple receiver beams) to angle-time envelopes of reverberation or single-target returns computed in BIREV to produce beam-time envelopes. Sums reverberant contributions of multiple volume-scattering layers and similarly can compute total received reverberation. Also manipulates beam-angle envelopes for multistatic geometries and multipulse transmissions.

The four support programs are:

- TLVSR – Computes and plots transmission loss vs range from the source or receiver to a boundary or a fixed target/layer depth along one or more bearings. Plots can be in either a 2D overlay or waterfall format.
- ACTENV – Scales surface, bottom, or volume reverberation and target-returns for a single receiver beam to absolute levels and overlays the results on a single 2D plot. Rather than processing target returns computed in BIREV, target returns can be calculated in ACTENV by using transmission losses computed in TLVSR.
- REVPLOTS – Scales reverberation and target-return beam-time functions computed in POSTBIREV to absolute levels and plots in a 3D format.
- ANGPLOTS – Scales reverberation and target-return angle-time functions computed in BIREV to absolute levels and plots in a 3D format.

The flowchart in Fig. 3 provides an annotated overview of the sequence of required program executions for a BiRASP prediction. Figures 4 to 7 illustrate the flow of required program executions for examples of specific analysis.

Although not restricted to such a configuration, BiRASP is designed primarily for a vertical line-array source and a horizontal line-array receiver. Further, BiRASP is designed for three generic applications: (1) a 3D bistatic geometry and a multiple-beam receiver; (2) a 3D bistatic geometry and a single receiver beam, for which an angular distribution of returns is calculated; (3) 2D monostatic or quasi-monostatic geometry and a single receiver beam, which is a retained capability of the earlier RASP model.

3D Bistatic Geometry and a Multiple-Beam Receiver

For this case, BiRASP applies an $N \times 2D$ approach over two sets of azimuthal sectors, one centered on each of the source and receiver locations and typically spanning 360° in azimuth. The sound-speed profiles and bathymetry within each azimuthal sector are assumed to vary only with range (constant over azimuth at any range) within the sector and are represented by 2D environment along a single azimuth, or radial, within the sector. A radial representation of an azimuthal sector centered on the source location is referred to as a source radial. A receiver radial is similarly defined. Depending upon the complexity of the environment, between 72 and 360 radials are recommended for each set of source or receiver radials. To facilitate the multiple executions of programs over radials, a UNIX shell script RASPLOOP is provided (see Section A5 of the Appendix).

For each source and receiver radial:

1. PROFIL and RAYACT are each executed once, and

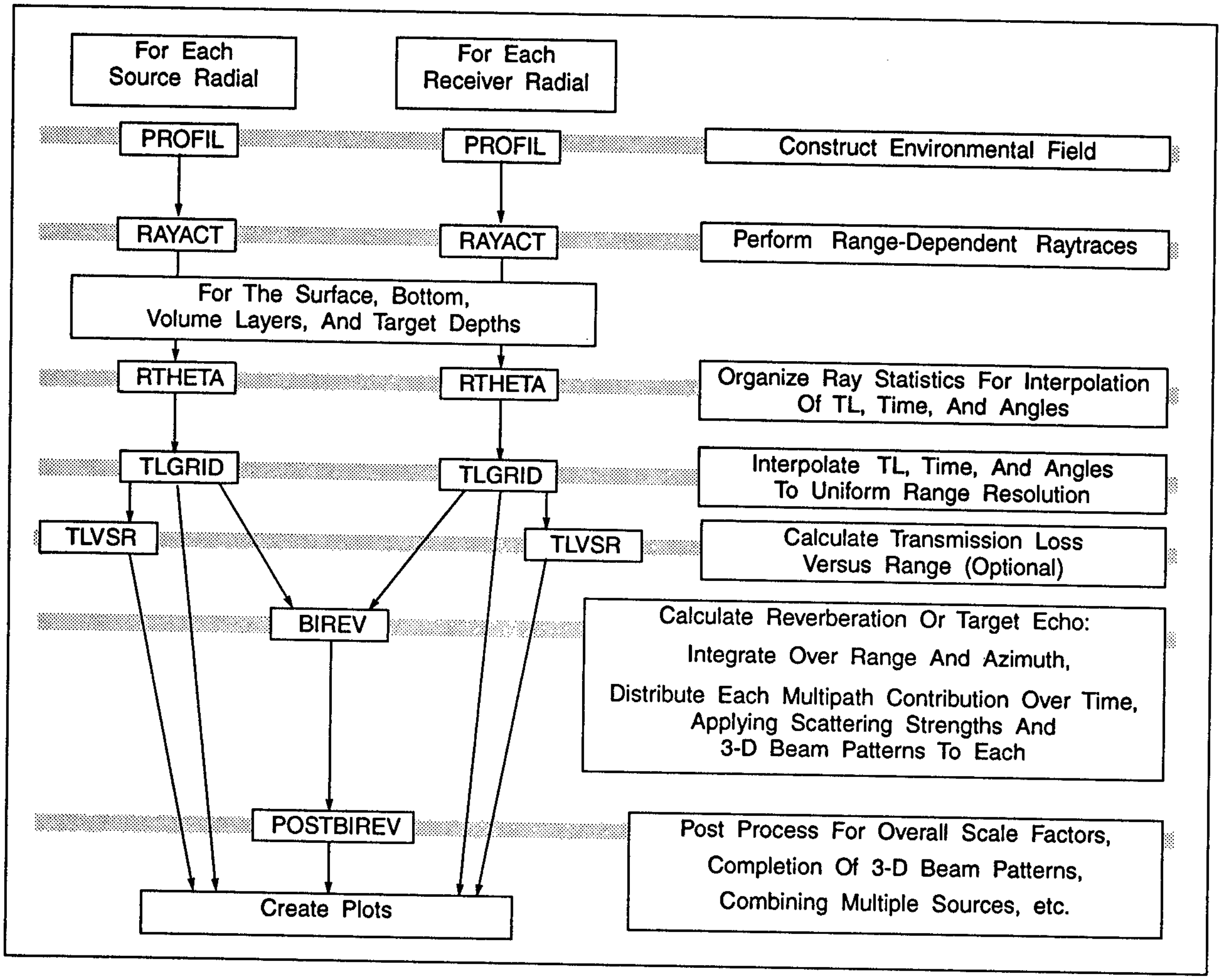


Fig. 3 – The sequence of required program executions for a BiRASP prediction

2. RTHETA is executed up to five times, once each for the
   (a) surface boundary,
   (b) bottom boundary, and
   (c) up to three depths, any of which may be the depth of a target or of a volume-scattering layer.

Program TLGRID is then used to process the multiple results of RTHETA corresponding to either all the source or all the receiver radials (azimuthal sectors) for one of the up-to-five boundaries and target/layer depths. Thus, TLGRID might be executed 10 times – once for each combination of the source or receiver with the 5 "depths." For each "depth," two TLGRID data files are generated: one containing RTHETA calculations for all the source-radials and the subject "depth," the other a similar data file of receiver-radial calculations. In each case, data are referenced to a polar coordinate system centered on either the source or the receiver location.

Next, program BIREV, which computes "raw" (uncalibrated) reverberation and target returns, further compresses the number of data files by combining source and receiver calculations of TLGRID. The program is executed separately for each boundary and target/layer depth. Thus BIREV

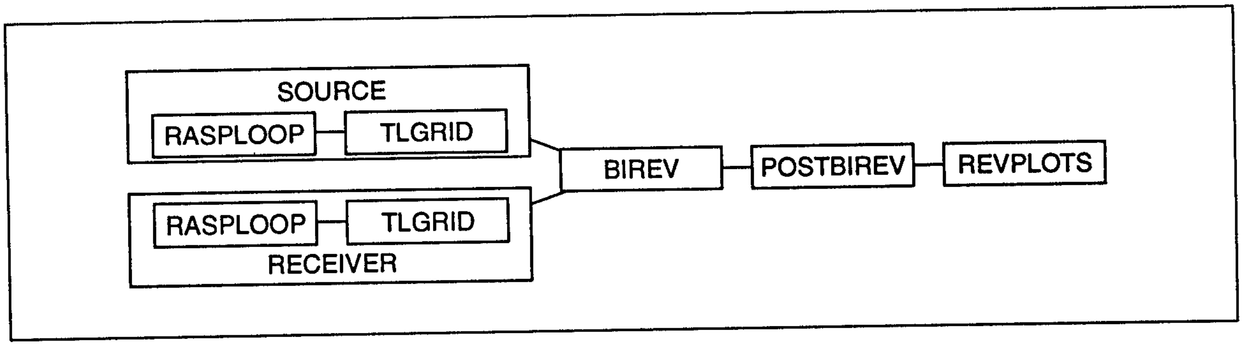


Fig. 4 – Required program executions for calculating multibeam reverberation

might be executed five times: once for each "depth." At this stage, returns are functions of time and the "array-raypath" angle between the array (baseline) and the returning raypath arrival. Further, reverberation from different scattering-layers has not been summed to estimate total volume reverberation.

Program POSTBIREV is then used to further process the results of BIREV. The multiple 3D beam patterns of the receiver are applied to produce beam-time series of returns. POSTBIREV can either be executed separately for each target-depth or source of reverberation (surface, bottom, and volume-layer) or sum sources of reverberation. For example, reverberation from individual scattering layers can be summed to yield total volume reverberation.

Program REVPLOTS scales reverberation or target-return beam-time envelopes computed in POSTBIREV to absolute levels and plots in a 3D format.

Figure 4 shows a flowchart for this case.

3D Bistatic Geometry and a Single Receiver Beam; Angular Distribution of Returns

As an analysis tool, BiRASP can compute returns as functions of time and some "nontraditional" angle, such as bistatic angle or scattering angle. This necessarily restricts the receiver to a single receive beam.

In this case, programs PROFIL, RAYACT, and RTHETA are executed in the same manner as for the case of a multiple-beam receiver, described above. In program BIREV, however, both the source and receiver beams are applied, and an angle (of distribution) is specified. At this stage, returns are functions of time and the specified angle parameter.

Program ANGPLOTS scales the angle-time envelopes computed in BIREV to absolute levels and plots in a 3D format. Program POSTBIREV is not executed in this case.

Figure 5 shows a flowchart for this case.

2D Monostatic or Quasi-Monostatic Geometry and a Single Receiver Beam

This "single-bearing" 2D case is a retention from the previous model RASP. In this case, only one radial and corresponding PROFIL execution is required. For a monostatic geometry (colo-

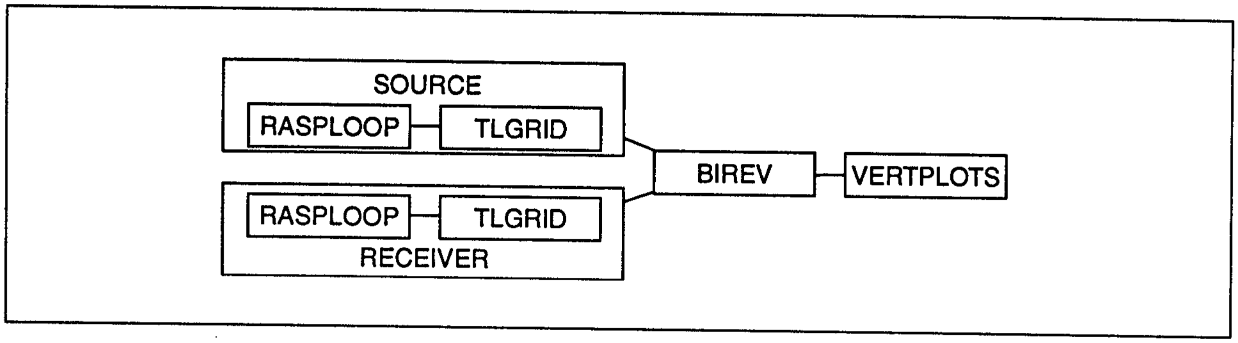


Fig. 5 – Required program executions for calculating the angular distribution of reverberation

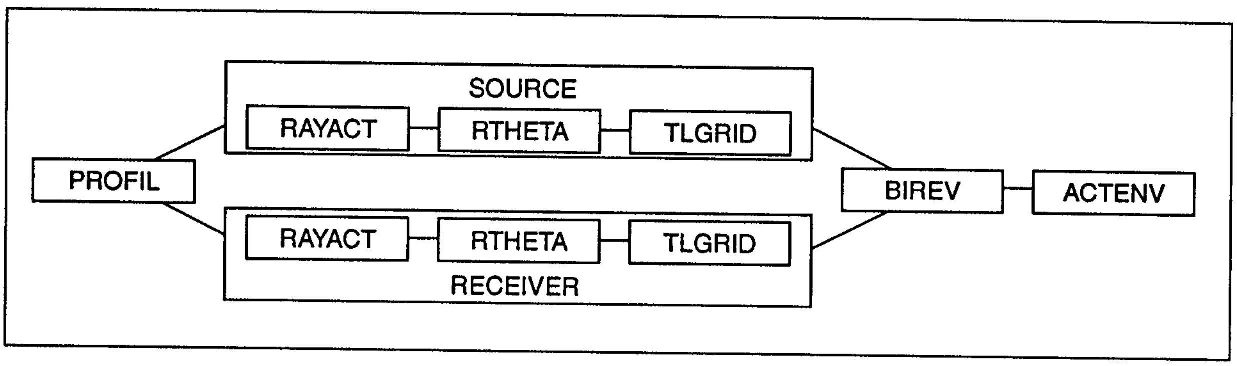


Fig. 6 – Required program executions for calculating reverberation along a single radial

cated source and receiver), programs RAYACT, RTHETA, and TLGRID are executed once for each boundary or target/layer depth. For a quasi-monostatic geometry (source and receiver separated in depth), the number of program executions doubles (hence are the same as for a bistatic geometry) because of the distinction of source- and receiver-related raypaths. Program TLGRID is executed separately for the source and receiver if they are separated in depth or possess different beam patterns. Vertical slices of beam patterns are applied in TLGRID.

BIREV is executed separately to compute surface, bottom, and volume-layer reverberation. Program ACTENV scales the 2D single-beam returns to absolute levels and overlays the results on a single 2D plot. Target returns, as a function of range, are calculated in ACTENV, using transmission losses previously computed in program TLVSR. Program TLVSR computes and plots total (vice individual raypath) transmission loss vs range from the source or receiver to a boundary or fixed target/layer depth, along a bearing.

Calculations for a "single-bearing" 2D geometry can proceed using TLGRID calculations for a single radial. Further, target returns can be calculated in BIREV, rather than in ACTENV.

Figure 6 shows a flowchart for this case.

Additional Comments

The modular structure of BiRASP allows for the re-execution of programs within the normal sequence of program executions. For example, additional volume-scattering layers are accounted for by additional executions programs RAYACT through BIREV and the application of program

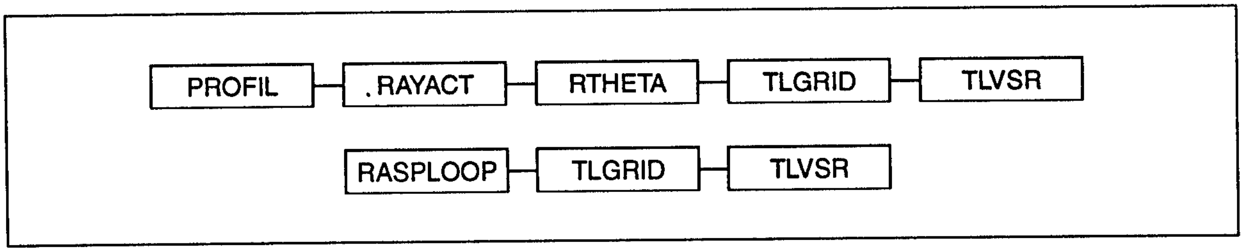


Fig. 7 – Required program executions for calculating transmission loss (one-way) vs range along a single radial (or along multiple radials using the UNIX script RASPLOOP)

POSTBIREV, which can sum and manipulate a number of envelopes. The modular structure also permits the user to change sonar parameters such as pulse-length and beam pattern without having to rerun the entire sequence of BiRASP programs.

Program TLVSR computes and plots total (vice individual raypath) transmission loss vs range from the source or receiver to a boundary or fixed target/layer depth along one or more bearings. Figure 7 shows a flowchart for calculating transmission loss vs range. TLVSR is an analysis tool not required by the BiRASP model. The ability to calculate target returns in ACTENV is a retention from the earlier RASP model. Rather than processing target returns computed in BIREV, target returns can be calculated in ACTENV using transmission losses previously computed in TLVSR.

The BiRASP model is written in standard FORTRAN. The source code is modular and extensively documented. The code is reasonably portable; it has been successfully compiled on the VAX Model 6310 computer and the SUN and SGI color workstations. The model's full capabilities are best realized on a computer that can perform memory swapping to disk. All user-supplied data are input via ANSI-standard, free-field READ statements, or ASCII files. Samples of the ASCII files are given with the input descriptions in the Appendix. Other than the optional print or log files, model-generated data files are either sequential or direct-access binary files.

The model uses the DISSPLA graphics package by Computer Associates for the majority of the black and white plots. It will be clearly indicated when a particular plot uses/requires this package. For users that do not have DISSPLA, a set of dummy subroutines are provided so that the programs will still compile. For the plots that do not use DISSPLA (e.g., all of the color plots), the format of the corresponding output file will be clearly specified in the documentation provided with the source-code distribution package. It should be a simple matter to adapt the output to the user's favorite plotting package.

The primary distribution package for the BiRASP model is configured for installation on a UNIX system. When uncompressed and unarchived, the package establishes the directory structure necessary for compilation. UNIX "makefiles" are provided for compiling each executable and are very detailed with respect to the source file dependencies. A variety of utility programs are also provided. They include programs to convert many of the binary files to ASCII for inspection and UNIX shell programs to manage the makefiles and crossreference the main files with the various "INCLUDE" files. In addition to sample input and output files, program notes are included that describe how the variable arrays may be adjusted when there is insufficient computer memory. The program notes also cover errata/modifications made to the input structure since this report was released. Altogether, the uncompressed and unarchived distribution package will require approximately 1.5 Mbytes of disk space.

4. PROGRAM DESCRIPTIONS

To exploit the versatility of the BiRASP model and to properly interpret its results, it is useful to understand how each program of the model performs its function. Therefore, a moderately detailed discussion of each program is provided in the following paragraphs. The sections describing the radial programs PROFIL, RAYACT, and RTHETA are essentially the same as those in Ref. 1. This is because the underlying algorithms have not changed. But since the differences involve changes in the programs' input structures, the reader is advised to refer to the current report.

4.1 Program PROFIL

Program PROFIL processes input sound-speed and bathymetry data to create a data file describing the 2D (range-depth) ocean environment along a radial track, for subsequent use by the ray-tracing program RAYACT. Input data are read from two user-specified files – one consisting of sound-speed profiles and their locations in range and the other of bathymetry data.

The discrete bathymetry data are connected by linear segments, from which bottom depth and slope as functions of range are calculated in program RAYACT. Each input sound-speed profile is corrected for the Earth's curvature and fit with a (two-prime) cubic spline fit. This curve fit is identical to that used by the GRASS ray-tracing model [12]. The purpose of the spline fit is to obtain continuity in the first two derivatives of sound speed, with respect to depth.

To incrementally iterate a ray path from a range-depth location, the ray-tracing algorithm in program RAYACT requires the sound speed, its first two derivatives with respect to depth, and its first derivative with respect to horizontal range at the range-depth location. This is accomplished by linearly interpolating and differencing values calculated for two range-adjacent sound-speed profiles that bracket the range of interest. The calculations for the sound-speed profiles are performed using their spline-fit representations.

The interpolation and differencing of values from the range-bracketing sound-speed profiles can be accomplished by RAYACT in either of two ways. The more common approach is to calculate the sound-speed and its first two depth derivatives at the depth of interest in the two profiles that bracket the range of interest. These values are then linearly interpolated in horizontal-range to estimate their values at the range-depth location of interest. The range-derivative is calculated from first differences of sound speed in horizontal-range.

A second available approach attempts to account for range dependence in the depths of surface ducts and deep sound channels. In this case, first the surface (zero) and maximum (made common to all profiles) depths of all the sound-speed profiles are connected by horizontal line segments. Next, the depths of the surface ducts (taken to be zero when a duct does not occur in a profile) are linearly connected between the range-distributed profiles. Thus for a profile that has a surface duct and for which the next (in range) profile does not, a line is drawn from the depth and range of the duct in the first profile to the sea surface at the range location of the second profile. This models the "disappearance" of the surface duct. Finally, the minimum sound-speed axis of up-to-two deep sound channels are connected between consecutive profiles to model, for example, a range dependence of the depth of the SOFAR channel. When the above "profile connections" are completed, the range-depth rectangle between each consecutive pair of sound-speed profiles will have been divided into trapezoidal regions. If requested, when running RAYACT, sound speed and its spatial derivatives will be computed between adjacent input sound-speed profiles by linear interpolation within the trapezoids. However, the automated connection procedure in PROFIL does not always produce the desired results; therefore the use of the profile connections for range-dependent sound speed in RAYACT should be used with caution.

In addition to creating print and data files of sound-speed and bathymetry information, PROFIL can, as an option, produce one or more plots of the environmental data. These plots are of:

- sound-speed contours on a range-depth grid (Note: requires DISSPLA),
- a composite plot of sound-speed profiles and bathymetry on a range-depth grid (e.g., Fig. 16 on page 41), and
- the individual sound-speed profiles showing how the spline fits to the input data.

Section A2 on page 51 describes the input data structure of program PROFIL.

4.2 Program RAYACT

RAYACT reads an environmental data file created by PROFIL and applies a 2D, range-dependent, ray-tracing algorithm. The algorithm is used to determine the ranges and associated ray statistics of ray-path encounters with the (flat) sea surface, the linearly segmented, range-dependent bathymetry, and up to three fixed target or volume-scattering-layer depths. The ray statistics include initial angle, grazing angle, travel time, number of ray cycles, and accumulated surface and bottom loss for each ray path. This information is recorded in data files (one for each boundary and target/layer depth for subsequent use by RTHETA). As an option, a plot of ray paths can be produced. The program does not compute ray intensities or (total) transmission loss. These quantities are calculated in RTHETA and TLVSR, respectively.

The source (receiver) depth, source frequency, and initial launch angles must be specified. Initial ray angles (measured positive-down from the horizontal) are specified by one or more non-overlapping ray-angle "fans", which are angular intervals of equally spaced and increasing angles (e.g., from $-14°$ to $+14°$ in steps of $0.2°$). The user can specify LONG RANGE DEFAULT, SHORT RANGE DEFAULT or a file of user-specified ray fans. The user-specified file can contain any number of ray fans as long as the total number of rays does not exceed 500, and the minimum allowable angular sampling resolution is $0.1°$. Table 1 gives the ray fan angles used by RAYACT for the two default cases. Essentially, the SHORT RANGE DEFAULT case provides higher resolution.

The selection of initial ray angles is somewhat subjective but improves with experience. A major criterion for the effective selection of initial angles is that the linear connections of the resulting samples of the range-angle contours of boundary and target/layer depth encounters adequately describe these contours (see the following discussion of program RTHETA). That is, linear interpolation providing good estimates of contour slope, especially in regions of main beam interactions and near caustics.

Snell's law can be used to estimate critical angles at the surface, bottom, and target depths. With this information and the angular extent of the vertical main beam (of the source or receiver), the user is advised to apply generally a relatively fine angular resolution (e.g., $0.2°$) for rays: (1) in a main beam, (2) that may encounter a boundary at a low grazing angle, or (3) that may intercept

Table 1 – Default Ray Fan Angles Used By Program RAYACT

Fan no.	Starting Launch Angle (°)	Final Launch Angle (°)	Angle Increment (°)
Ray Fan Angles for LONG RANGE DEFAULT			
1	−84	−33	3.0
2	−30	−23	1.0
3	−22	−15	0.5
4	−14	+14	0.2
5	+15	+22	0.5
6	+23	+30	1.0
7	+33	+84	3.0
Ray Fan Angles for SHORT RANGE DEFAULT			
1	−88	−30	1.0
2	−29.5	−20	0.5
3	−19.8	−10	0.2
4	−9.9	+9.9	0.1
5	+10	+19.8	0.1
6	+20	+29.5	0.5
7	+30	+89	1.0
Note: angles measured positive-down from the horizontal.			

a target depth at a low angle. The latter two cases generally correspond to approximately ±14° in the deep ocean. The angular sampling resolution can then be gradually decreased for the steeper initial angles.

Ray trajectories can be terminated in a number of ways. These include exceeding maxima for any one of the following:

- propagation range,
- one-way travel time,
- number of bottom reflections,
- accumulated surface/bottom loss, or
- ray-order (see program RTHETA).

The user can specify the above values, or allow the program to assign default values, or compute values from available information (See Section A3 on page 53 for details). Also, the maximum ray order to process can be specified later when running RTHETA.

Reference 13 describes the method of ray trace used in RAYACT. The procedure is an iterative one in which a ray is incremented from point to point along its path. This is accomplished by evaluating Taylor series expansions in arc length of various ray parameters such as range, depth, travel time, and ray angle. These expressions are derived from the basic ray equation:

$$\frac{d}{ds}\left[\frac{1}{c(\rho,z)}\,\frac{dP}{ds}\right] = \nabla\left[\frac{1}{c(\rho,z)}\right] . \qquad (10)$$

The sound speed $c(\rho, z)$ is assumed to be known at every range ρ and depth z along the 2D medium. P is the positional vector to a point on the ray, and s is arc length along the ray.

The iterative computation of ray trajectories requires the specification of certain iteration step-size and accuracy-test parameters. Again, the user can specify the values of these parameters or allow the program to assign default values. It is recommended that the user allow these default values to be used except for special cases where extreme accuracy is sought. Reference 13 provides additional details of the iterative ray-trace algorithm.

Surface and bottom reflection loss functions are also input in RAYACT for up to 50 range intervals. The user is prompted for the name of a file containing the bottom loss values.

Alternatively, RAYACT has bottom-reflection loss functions internally stored. These functions can be specified by identifying FNWC bottom types. A user may input up to five FNWC bottom types, ranging from type 1 to type 5, along with the maximum range for this bottom. It is important to note that the bottom-loss functions for bottom types 1 and 2 are identical, as are those for bottom types 4 and 5 [5]. Bottom type 1 is typically a low-loss bottom, while bottom type 5 is typically a high-loss bottom.

Program RAYACT generates output in several forms. The user may select any combination of the following:

- data files of surface, bottom, and target/layer depth ray calculations for input to RTHETA,
- a print file of detailed ray boundary and target/layer depth encounter information (selected rays), and
- a plot of selected ray paths (e.g., Fig. 17 on page 41, Note: requires DISSPLA).

The input data structure of program RAYACT is described in Section A3 on page 53.

4.3 Program RTHETA

The function of RTHETA is to reorganize the ray-tracing results of RAYACT into a form (order contours) amenable to calculating transmission loss and reverberation. RTHETA also calculates the transmission loss for each ray path encounter with a boundary or target/layer depth. A single execution of RTHETA processes only one boundary or target depth and generates an output file that will be processed by the program TLGRID.

The concept of order contours is illustrated for the cyclic ray propagated from the point S/R in Fig. 8. For numbering purposes, each ray path is divided into segments bracketed by reversals in ray trajectories due to boundary encounters or turns: segments that terminate with a ray crest or surface encounter are numbered with odd integers; segments that terminate with a ray valley or bottom encounter are assigned even integers. The integers increase with distance from the source and are used to classify the state of the ray by the number of oscillations it has made. Ray paths with an equal number of oscillations are said to be of equal order. The order contours for boundary interactions are derived from the range at which rays encounter a boundary, whereas for target or layer depths, they are derived from the range at which rays cross the target/layer depth. In Fig. 9, order contours are shown as curves determined by the encounters (boundary or target/layer depth) of a given order and plotted on an initial source angle vs range coordinate system. A given contour

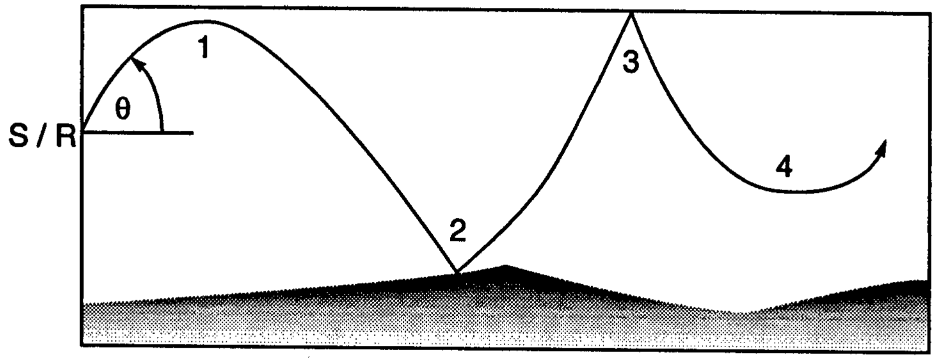


Fig. 8 – Numbering of ray crests, ray valleys, and boundary encounters illustrated for one ray

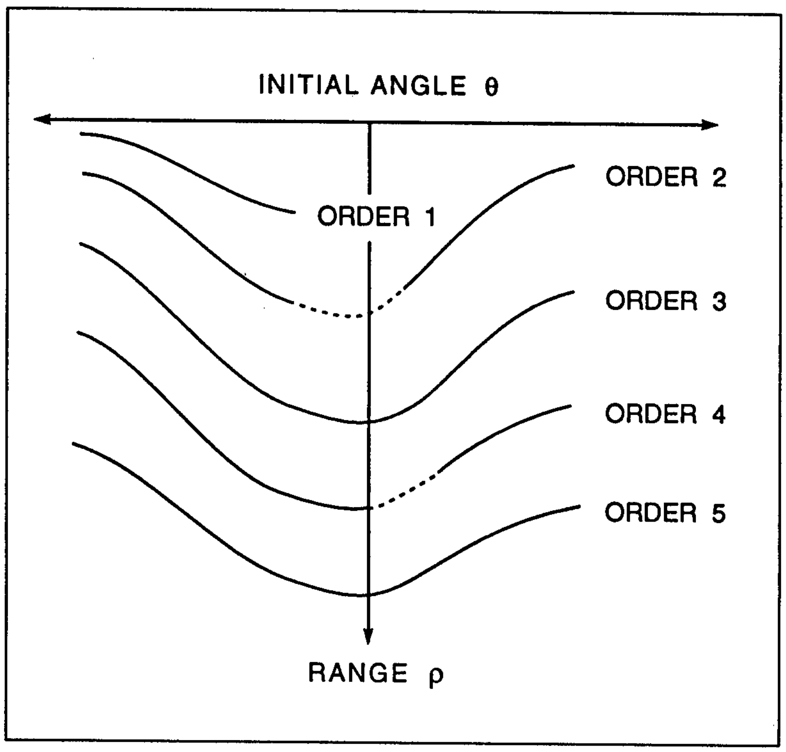


Fig. 9 – Order contours for either a single target/layer depth or for both boundaries. Dashed sections correspond to intervals of ray reversals that did not encounter a boundary or cross a target/layer depth.

need not necessarily be a continuous curve but may consist of several disjoint segments separated by intervals of ray turning points, which cause the rays to not encounter a boundary or to not cross a target/layer depth.

For a target/layer depth, all of the order contours are used, whereas for a given boundary, only those contours corresponding to that boundary are pertinent; they are odd for the surface and even for the bottom. For example, to determine all ray paths between the source point and the bottom point (x, y) at range ρ, consider a horizontal slice at ρ, as Fig. 10 shows. If the three paths having initial angles θ_1, θ_2, θ_3, denoted by 1, 2, 3, respectively, then the index set $m(x, y)$ of Eq. (1) consists of the integers 1, 2, and 3.

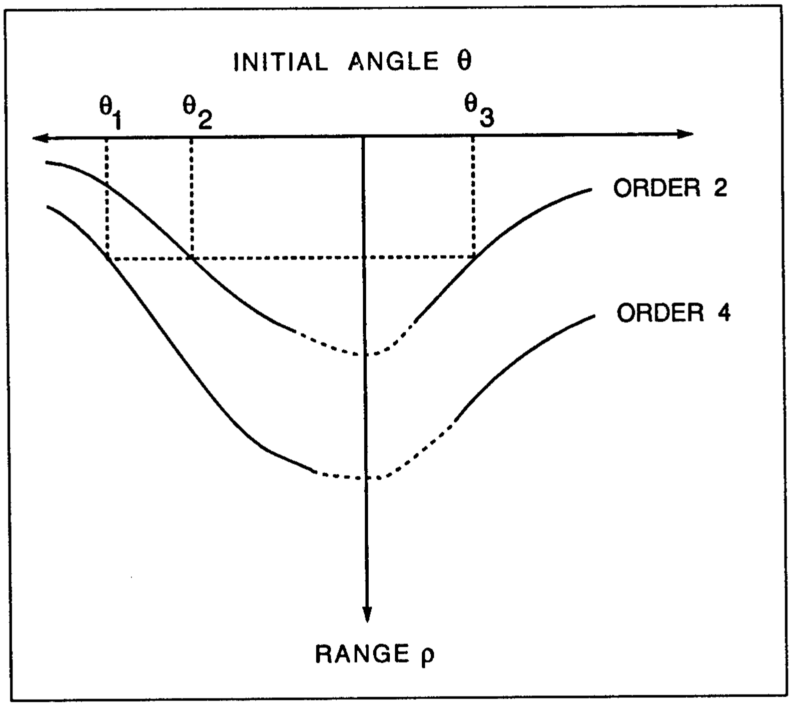


Fig. 10 – Calculation of launch angles, i.e., rays that meet bottom boundary (curves of even order) at range ρ

The finite number of rays traced produces discrete samples of an order contour. Figure 11 illustrates the construction of two order contours of the type produced by the RTHETA program for bottom interactions. Here Bs represent the bottom encounters and the Vs represent the refracted turning points (valleys) determined by tracing rays with initial angles $\theta_1, \theta_2, \ldots, \theta_n$. The order contours are approximated by linearly connected, consecutively computed, bottom encounters of the same order. For contour points between the computed points, the travel time, transmission loss, and the angle at which a ray encounters a boundary can be found by interpolating the corresponding computed values linearly with respect to range. The use of ray orders thus facilitates the determination of all the ray paths connecting the source-receiver point S/R with a boundary element at any range ρ.

Figure 12 illustrates a smoothed surface order contour obtained for a typical deep-ocean sound-speed profile. The two rays that just graze the bottom (corresponding to upward and downward initial angles) determine points B and B' on the contour. The two rays that graze the surface occur at shallower angles. These rays contribute points A and A' on the contour. Thus, rays corresponding to initial angles in the intervals (θ'_B, θ'_A) and (θ_A, θ_B) encounter the surface but not the bottom. These rays provide significant contributions to surface reverberation at long ranges. Rays having steeper initial angles encounter the bottom and thereby suffer bottom loss. The cumulative loss associated with several bottom reflections causes these rays to have less effect on surface reverberation at long ranges where the higher order contours are relevant.

Transmission loss is determined for points on an order contour by computing individual-ray transmission loss at the ray-encounter locations found by raytracing and applying linear interpolation between these points. Here, transmission loss consists of geometric spreading, absorption loss, surface/bottom loss, and weighting by a vertical beam pattern. Whereas surface and bottom loss

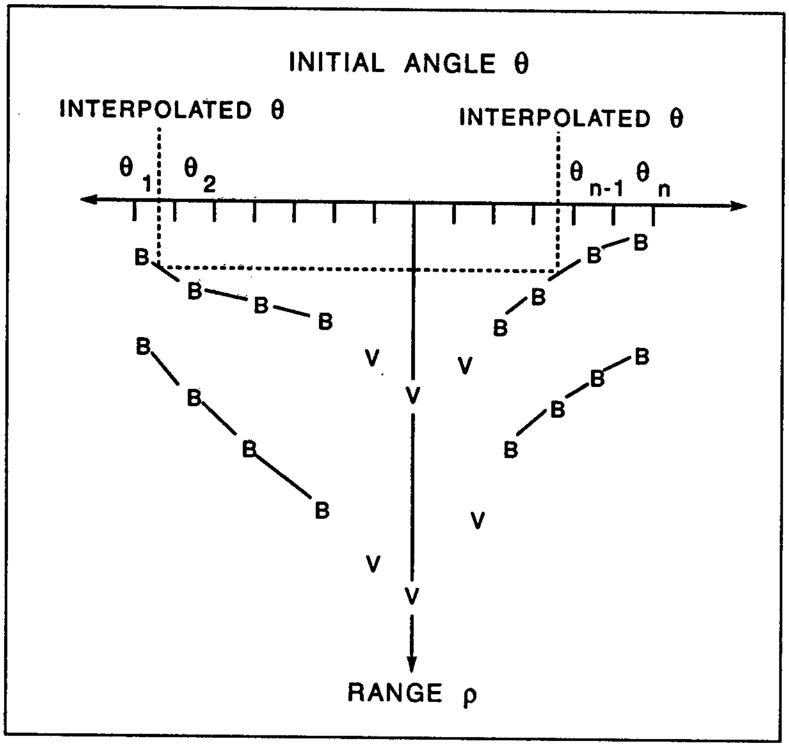


Fig. 11 – Computer construction of order contours $R(\theta, \rho)$

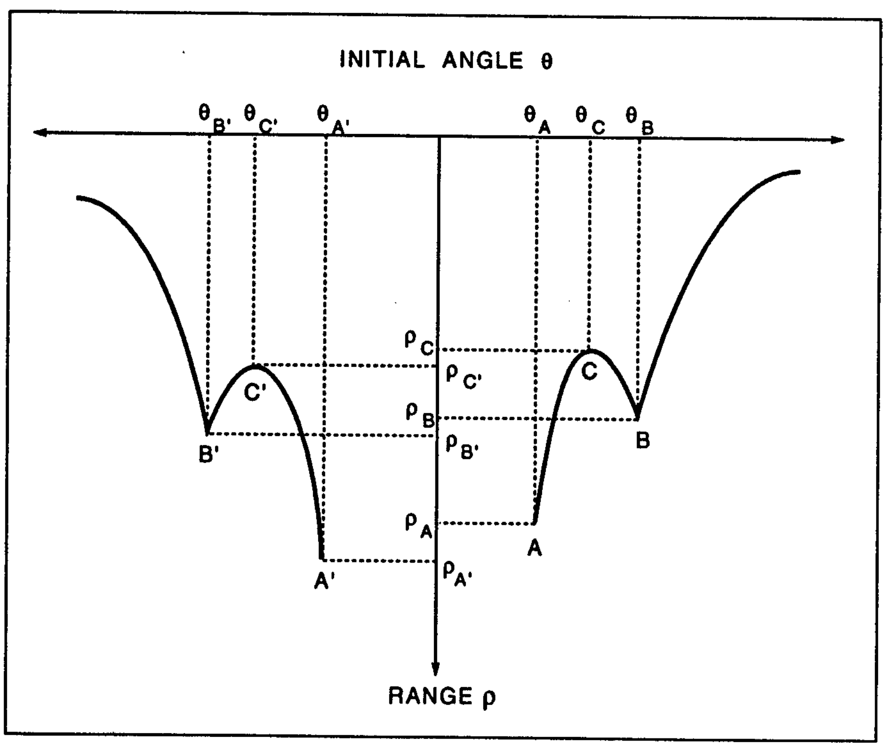


Fig. 12 – Smoothed surface order contour illustrating caustic behavior

are computed during ray tracing, the remaining losses are calculated after the order contours are constructed. Volume absorption is estimated by using Thorp's equation [14]. Geometric spreading loss under our assumption of azimuthal symmetry is given by:

$$L(\rho) = \left| \frac{\rho}{a^2} \frac{\sin\gamma}{\cos\theta} \frac{\partial\rho}{\partial\theta} \right| , \tag{11}$$

where a is the unit reference distance, ρ is the horizontal range, θ is the initial source angle of the ray, and γ is the angle the ray makes with the horizontal at range ρ.

The derivative $\partial\rho/\partial\theta$ in Eq. (11) is approximated numerically at each point (ρ_i, θ_i) on a contour. If θ_{i-1}, θ_i, and θ_{i+1}, are the successive initial angles of rays that are traced on this contour, and ρ_{i-1}, ρ_i, and ρ_{i+1} are the corresponding ranges, then

$$\frac{\Delta\rho_i}{\Delta\theta_i} = \frac{\Delta R_i}{\Delta R} \frac{\Delta R_{i-1}}{\theta_i - \theta_{i-1}} + \frac{\Delta R_{i-1}}{\Delta R} \frac{\Delta R_i}{\theta_{i+1} - \theta_i} , \tag{12}$$

where

$$\Delta R_{i-1} = | \rho_i - \rho_{i-1} | , \tag{13}$$

$$\Delta R_i = | \rho_{i+1} - \rho_i | , \text{ and} \tag{14}$$

$$\Delta R = \Delta R_{i-1} + \Delta R_i . \tag{15}$$

In using a ray-theoretic approach to modeling sound propagation, a serious deficiency arises when attempting to calculate geometric spreading loss in near-caustic regions – that is, near locations where adjacent ray paths cross. In this region, geometric spreading tends to zero, erroneously implying infinite intensity at the point of intersection. This phenomenon occurs when $\partial\rho/\partial\theta$ is equal to zero, or as related to order contours, where the contour of order n, $\rho = f_n(\theta)$ has a stationary value. The surface order contour shown in Fig. 12 has caustics at points C and C'.

Program RTHETA determines the locations of potential caustics ($\partial\rho/\partial\theta = 0$) and attempts to distinguish between real and false caustics. The latter are often generated by irregular bathymetry or at slope discontinuities of an order contour corresponding to transitions between ray-path "types." For example, the false caustics at points B and B' of Fig. 12 separate ray paths that have been bottom-reflected from those that have been refracted away from the bottom. For a real caustic, RTHETA applies a wave-theoretic smooth caustic correction in the region of the caustic. The particular formulation agrees with Spofford [15]. When the caustic correction is applied, additional discrete ray encounters are created to provide more accurate interpolation of intensity within the caustic region.

A practical problem arises when rays are traced to long ranges over irregular bottom topography or when a significant number of ray paths encounter a seamount or continental rise. In these cases, the corresponding order contours may tend to become highly irregular, with numerous false caustics. These contour irregularities can hamper the estimation of range derivatives needed for transmission loss calculations. To combat this problem, a statistical ray-averaging procedure has been developed that operates on an order contour and effectively smooths the irregularities in the transmission loss estimates. The procedure does not apply a caustic correction. Rather, it involves unfolding the order contour and providing an avenue around zeros in $\partial\rho/\partial\theta$. The procedure, developed by Palmer [16], performs ray-intensity averaging as a function of range.

Therefore, program RTHETA uses two methods to compute the transmission loss of individual rays – one that uses geometric spreading with a caustic correction (called a ray bundle calculation) and another that involves statistical ray-averaging. Only one of these types of calculations can be written to an output file for subsequent use by TLGRID, and this choice is user-specified. Normally, the ray bundle calculations are written to the output file. However, in cases of irregular bathymetry, it is recommended that the statistical averages be selected for contours of bottom encounters.

Occasionally, a ray will be erroneously traced in the sense that its cyclic behavior differs radically from the ray paths of neighboring initial launch angles. For example, when a source is placed below a shallow surface duct, a ray may inadvertently become trapped in the duct although its immediately neighboring rays are not. Such an occurrence can significantly distort the resulting order contours. To overcome such a situation, the user has the option to delete specific ray paths from the construction of order contours.

Upon completion, RTHETA will output

- a plot of order contours (e.g., Figs. 18 - 20 on pages 42 through 43, Note: requires DISSPLA),
- calculations of order contours and associated transmission losses on a data file for input to program TLGRID, and
- a print file of data describing the order contour curves.

The input data structure of program RTHETA is described in Section A4 on page 57.

4.4 Program TLGRID

Programs PROFIL, RAYACT, and RTHETA perform calculations along a single radial, which represents an azimuthal sector. For the general bistatic geometry (source and receiver separated in range), these programs are executed separately for each member of two sets of azimuthal sectors (or radials): a set of "source radials" that defines the 3D geometry centered at the source location and a set of "receiver radials" that defines the 3D geometry centered at the receiver.

Program TLGRID processes the multiple results of RTHETA corresponding to either all the source or all the receiver radials (azimuthal sectors) for one of the up to five boundaries and target/layer depths. The user must specify an input data file that contains a list of (radial) bearings and the file names of the corresponding RTHETA output files to be processed. Program TLGRID processes the appropriate output files of RTHETA by interpolating order-contour calculations onto to a sectored polar coordinate system with a uniformly-spaced range grid, primarily for subsequent use by program BIREV. The user specifies the origin of the polar coordinate system (which is either the source or receiver location) in latitude and longitude. The location of the origin is required by program BIREV to determine the bistatic geometry (e.g., source-receiver separation in range).

The use of 72 to 360 radials is recommended, providing azimuthal sampling at 1°–5° intervals, with the density chosen depending upon the complexity of the environment. The user specifies the name of the file that contains a list of bearings and the corresponding RTHETA file names. This input file is automatically created if the UNIX shell script RASPLOOP is used.

The interpolations in range performed in TLGRID are for ranges $\rho_n = (n - 1/2)\Delta\rho$ ($n = 1, 2, 3, \ldots$) out to the maximum range available for a given contour. The range-sampling interval

Δr must be small enough that the multipath structure over the interval does not change appreciably from that found at its midpoint. Also, the range-resolving capabilities of the acoustic pulse and signal processing must be taken into consideration, otherwise erroneous reverberant dropouts (times of no received reverberation) may result when the integrations are performed. The later constraint can be expressed by

$$\Delta\rho < \frac{c\,D}{2}\,, \tag{16}$$

where c is the sound speed and D is the pulse duration of the transmitted signal in seconds. A simple approximation to Eq. (16) is to equate $\Delta\rho$ to D where the latter is interpreted as being expressed in units of kilometers. Note that for complex waveforms, such as an FM sweep (which is processed by pulse compression matched filtering), the duration D is the pulse length of the equivalent continuous wave (CW) pulse, i.e., $1/W$ where W is the frequency bandwidth (Hz) of the transmitted waveform.

It is possible in TLGRID to apply a source or receiver beam pattern to the raypaths as each RTHETA file is processed. However, although it is computationally more efficient to apply beam patterns in TLGRID rather than BIREV, it is not generally recommended. The application of a beam pattern in TLGRID limits the user to a single beam pattern for either the source or the receiver. One of the strengths of the BiRASP model is its ability to calculate beam data for multiple (horizontal) beam receiver-systems; that is, beam output for several beams, each steered to a different azimuth or heading. Even for a single-beam source or receiver system it is generally best to apply the pattern in BIREV. Beam pattern applications by BIREV are generally far more accurate than those by TLGRID, especially in the case of only a few radials describing the environment. A vertical line array with no tilt (azimuthally symmetric beam pattern) is the exception. Finally, it is advantageous to delay the application of beam patterns as long as possible in the sequence of calculations to provide the flexibility to change beam patterns with a minimum amount of recalculations. Beam pattern calculations performed by BiRASP are discussed in detail in Section 4.5.2.

For each radial, after all the interpolated values of acoustic intensity, travel time, grazing angle, and launch angle have been accumulated, the "ray arrivals" in each range bin are sorted and down-selected. The user may sort the arrivals based on ray intensity, or ray intensity weighted by the beam pattern of a vertical array or weighted by the sine of the grazing angle, or weighted by both. The number of sorted "arrivals" to be written to the output file must be specified. Although the weighting by the sine of the grazing angle is only used for sorting purposes, the user can specify if the beam-pattern weighting is included in the output.

The primary purpose of the sorting and selection criteria is to anticipate the probable magnitude of an arrival's contribution to reverberation (or target echo) and keep only the most significant arrivals. In particular, the integration optimization algorithm (discussed in Section 4.5.4) exploits the decreasing magnitude of the contributions to minimize the execution time. A secondary purpose is to reduce the size of the output file, which is directly proportional to the number of radials, the number of range bins, and the number of arrivals per bin.

The input data structure of program TLGRID are described in Section A6 on page 62. The output of TLGRID consists of a single, direct-access file.

4.5 Program BIREV

The primary purpose of BIREV is to evaluate the integrals for boundary and volume reverberation and for target echo, Eqs. (5), (7), and (8), respectively. Its capabilities, in addition to allowing arbitrary source/receiver geometries, 3D beam patterns, and range-dependent bistatic scattering strength functions, include the ability to control most aspects of integration to:

- calculate reverberation as a function of some angle (e.g., receive angle, boundary grazing-angle, etc.) as well as time,
- limit calculations to the strongest individual raypath contributions, and
- minimize execution time.

Although BIREV is flexible with a wide range of possible analysis products, the program is primarily configured for a vertical-line, single-beam, source array and a horizontal-line, multiple-beam, receive array. Some of the design considerations concerned memory and disk space usage. Others were initially implemented to reduce execution time on older computer systems. They have been maintained for their analysis capabilities.

4.5.1 General Integration Procedure

Figure 13 shows the procedure used in the area integration of Eqs. (5) or (7). The area of integration is gridded in a polar coordinate system centered on the receiver in order to make efficient use of tabular files generated by TLGRID. A single azimuthal receiver sector is shown which will contribute to the reverberation field over the time interval of interest. This region of space is then divided into polar scattering subareas of radial increment Δr and azimuth increment $\Delta\phi$. The number of azimuth increments into which the sector is actually divided depends on the range to the receiver. The program sets $\Delta\phi$ so that the area of each of the scattering "cells" is, at most, $(\Delta r)^2$, and the angular resolution with respect to the receiver is less than 0.5°. For each azimuthal subarea, the range and bearing from the source position is determined. The transmission loss, travel time, etc., from the source to the scattering cell and from the scattering cell to the receiver for all ray arrivals are then retrieved from the lookup tables. The reverberant contribution for each possible pairing of "incident" source-rays with "scattered" receiver-rays is calculated and added to the reverberation time series at the appropriate two-way travel time. The calculation is repeated in turn for the rest of the range sectors until the maximum range of interest is reached.

There are numerous inputs associated with controlling the integration.

<u>Coordinate System:</u> Calculations can be performed using a Cartesian coordinate (planar) system, or they can take the Earth's curvature into account, using Great Circles and a spherical coordinate system. The Cartesian system is based on a Mercador projection about the receiver's location. For either coordinate system, absolute azimuthal angles or bearings are measured positive, clockwise from North.

<u>Radial Sector Assignments:</u> Since it is not computationally practical to execute a raytrace to every scattering cell, the user must specify the limits of the azimuthal sector to be associated with each radial. The user specifies, for each radial from the source (receiver), a range of angles about the radial's bearing within which that radial's acoustic information will be considered valid. This range of angles is called that radial's *sector*. The sectors for the source (receiver) should not overlap. The sectoring can be done automatically or manually. Automatic sectoring

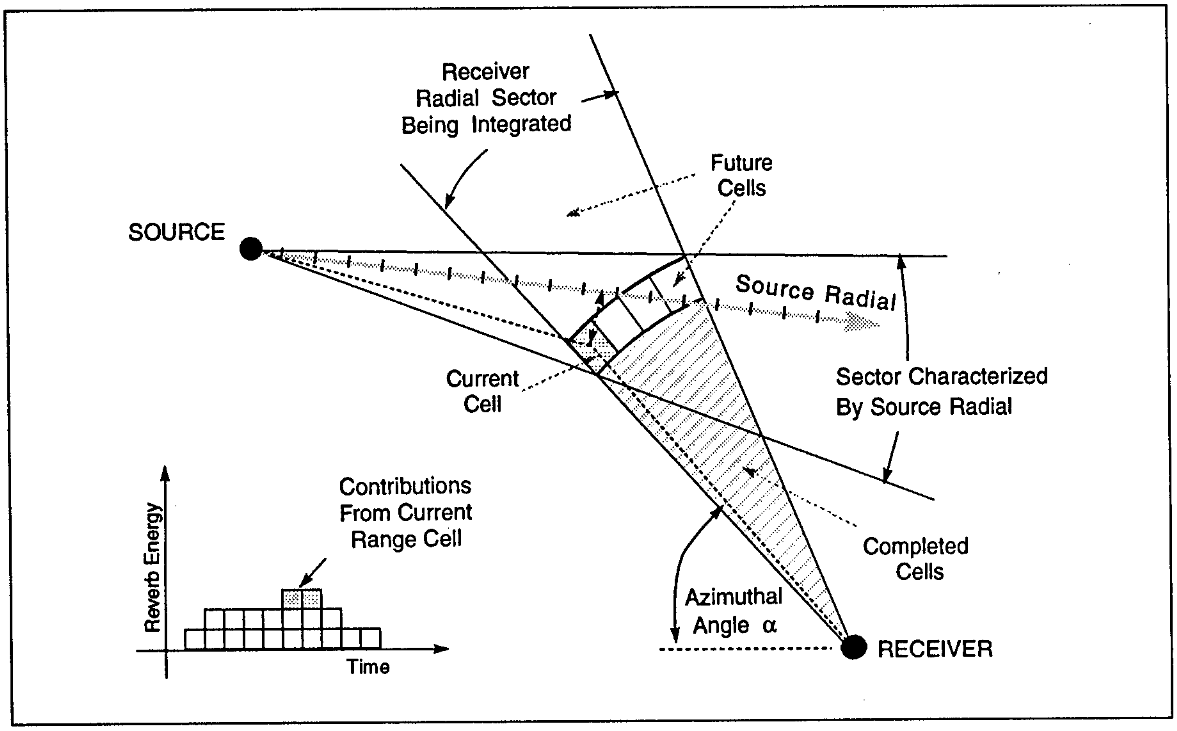


Fig. 13 – Integration procedure for calculating reverberation due to scattering along an angular sector. The angle α measures the direction of arrival of the reverberation relative to an arbitrary coordinate reference.

means that the program will determine each radial's sector using the following algorithm: the beginning of a radial's sector is defined as the bearing located half the angular distance from the radial's bearing to the bearing of the radial immediately counterclockwise; the end of a radial's sector is defined as the bearing located half the angular distance from that radial's bearing to the bearing of the radial immediately clockwise. Manual sectoring means that the user specifies the angular range of each radial's sector.

Angular Ranges of Integration: For both the source and receiver arrays, a range of azimuthal angles must be specified to define the area that can contribute to the reverberation. If the user does not want to limit the area around an array, a default angular range of 360° may be specified. This ensures that the entire area surrounding that array can contribute to the reverberation. If the user selects the angular ranges of integration, the program checks for inconsistencies. Limiting the area of integration allows the left-right ambiguity of horizontal line arrays to be resolved.

Multipaths Used: In TLGRID, raypath arrivals at each range bin of a given radial (azimuthal sector) are sorted by an estimated strength of their potential contribution to a reverberant or target return. The arrivals are numbered from 1 (for the strongest) to some number (specified by the user) for the weakest saved. In BIREV, the user must specify, for both source and receiver, a subset of the sorted arrivals that will be included in the calculations. For example, to limit calculations to the ten strongest arrivals (at a range bin) from each the source and receiver, the user specifies both the source and receiver subsets as ranging from 1 to 10. When used in conjunction with the sorting capabilities of TLGRID, the ability to control which multipaths from a source radial and from a receiver radial are to be paired allows the user to assess the impact that different pairing combinations have on the final prediction; e.g., which multipaths are the most important contributors to the reverberation.

Pulse Duration & Time Resolution: The transmit signal is specified by its duration or pulse length (s). For CW and broadband noise signals, the definition of pulse length is simple. However, for complex waveforms, such as those processed by pulse-compression matched filtering (e.g., a frequency modulated (FM) sweep), the duration should be the pulse length of the equivalent CW pulse, i.e., $1/W$, where W is the frequency bandwidth (Hz) of the transmitted waveform.

The reverberation $R(t_l)$ calculated for time t_l is not instantaneous power. Rather, an average power calculated by dividing the reverberant energy received over the Δt time interval preceding t_l by Δt. For small Δt on the order of a second, the difference between instantaneous power and average power should be insignificant. Larger Δt will act to smooth the reverberation envelope with Δt corresponding to an averaging time.

When calculating the general area assessment of target echo, the program assumes there is is target at each scattering patch used to calculate reverberation, and only one value is recorded for that patch. If the time resolution is too fine, then there are many more time bins than targets, which results in plots with numerous target level dropouts. To prevent this, the time resolution should be set to approximately $\Delta r/(2c)$, where Δr is the range sampling rate used in TLGRID.

Angular Distribution of Reverberation: Typically, BiRASP is used to calculate returns received on the multiple beams of a horizontal line array. In this case, BIREV is used to calculate returns as a function of time and the "array-raypath" angle between the array (baseline) and the returning raypath arrival. The multiple 3D beam patterns of the receiver are then applied in program POSTBIREV. Another common approach is to calculate returns as a function of azimuth with respect to an omnidirectional sensor or vertical array to geographically locate sources of reverberation. As an alternative to these more standard calculations, BIREV allows the user to calculate returns as a function of some other angular parameter. Table 2 shows the available parameters. The default resolutions may be changed via FORTRAN PARAMETER statements. The choice of angle parameter, if any, affects the application of beam patterns. Except when the array-raypath angle is chosen, all source and receiver beam patterns must be applied in BIREV or TLGRID. When the array-raypath angle is chosen, the array may be either the source or receiver, and the beam pattern for the sensor not selected must be applied in BIREV or TLGRID.

Direct Blast: If the source/receiver geometry is bistatic, then it is possible to include the direct-blast component of reverberation in the prediction, i.e., the reverberation directed along the line connecting the source and receiver. To include direct-blast in the prediction, the user must specify the name of the TLGRID file that contains acoustic information for propagation from the source directly to the receiver. This file should have the same beam-pattern processing as applied to the other source TLGRID files.

Single-Bearing Calculations: For cases when the source/receiver geometry is monostatic or quasi-monostatic and it is possible to impose azimuthal symmetry, the user may select a single bearing from the TLGRID files and perform the integration only over range, i.e., $2\pi \int(\cdots)\rho d\rho$. The beam patterns that are applied correspond to a vertical slice for the selected bearing through the 3D pattern of the array. For predictions that involve a vertical or near-vertical source array, a near-broadside beam for a horizontal receiver, and negligible sidelobe contamination, this approach can result in significant savings in execution time. The reverberation levels for a horizontal receiver should be reduced by the Receiver Directivity Index (RDI $= 10\log_{10} N$, where N is the number of elements in the receiver array). Note that when the user selects this

Table 2 – Summary of Angles for which of Reverberation may be Calculated

	Minimum Angle (°)	Maximum Angle (°)	Default Resolution (°)
None	NA	NA	NA
Angle between a source or receiver ray and the corresponding line array (array-raypath angle)	0.0	180.0	0.5
Azimuthal angle wrt the source or the receiver	0.0	360.0	1.0
Vertical Launch (Arrival) angle wrt the source (receiver)	-90.0	90.0	0.5
Bistatic angle (angle between the bearings from the scatterer to the source and receiver)	0.0	180.0	0.5
Incident or Scattered grazing angle at the boundary	0.0	90.0	0.25
Incident or Scattered angle from a volume layer	-90.0	90.0	0.5
Average of the incident and scattered angle from the boundary	0.0	90.0	0.25
Arcsine of the geometric mean of the sines of the incident and scattered angle from the boundary	0.0	90.0	0.25

option, various inputs or options are not required (i.e., Radial Sector Assignments and Angular Ranges of Integration).

Optimization of Area Integration: The execution time for a BIREV reverberation prediction may be significantly decreased by using an optimization algorithm. In brief, the algorithm shortens the time spent pairing source and receiver boundary encounters from a scattering element by terminating the calculation when the contribution to the reverberation from all remaining pairings is negligible in comparison with the contribution from all previous pairings. The user specifies a relative error factor ϵ between zero and one. The accuracy is higher the closer ϵ is to zero. Inputting a value of zero for ϵ results in all possible pairings being calculated. The optimization algorithm is not available for single-bearing calculations. A full description of the optimization algorithm is given in Section 4.5.4.

4.5.2 Beam Patterns

BIREV allows for fully 3D beam patterns to be applied to both the transmitted and received signals. The program routinely provides for sources and receivers that are either omnidirectional or uniformly spaced linear arrays (with spatial shading). The physical orientation of both arrays is always required. Thus for each array, the user must specify:

- The array's primary configuration, i.e., omnidirectional, horizontal, or vertical.
- The array's heading in degrees, measured positive clockwise from north.
- The tilt of the array, specifying the deviation in degrees from the appropriate axis. Figure 14 shows the conventions for the arrays' tilts.

If during the execution of BIREV(as opposed to POSTBIREV), an array is to be beamformed or steered, the user will also be required to specify:

- The total number of elements and the total number and index (element number) of any dead (inoperative) elements. Dead elements will be assigned a weight of zero.

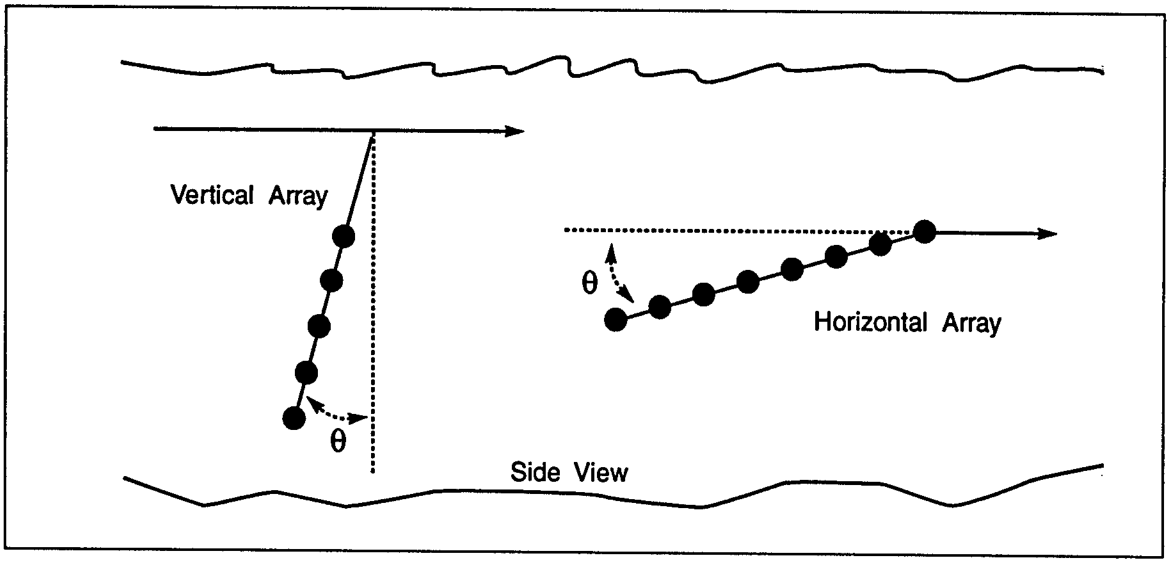


Fig. 14 – Conventions for specifying the positive tilt θ of horizontal and vertical linear arrays. Shown for each are the vertical plane containing the array, the solid horizontal lines that indicate the heading of the array, and the dotted reference lines for determining θ. An array's orientation specified using "pitch and roll" must be converted to this convention.

- The physical spacing of the elements and the spatial shading to be applied to the elements. Hamming, Hanning, or uniform shading may be selected. Alternatively, a user supplied file of weights may be used.
- The direction of the main response axis of an array may be specified via a steering angle or a beam number. The conventions for specifying steering angles are shown in Fig. 15. Numbered beams are uniformly spaced in cosine space and span 180 degrees. The direction of beam 0 must be specified (e.g., forward or aft for horizontal arrays).
- The actual speed of sound at the array center, and the speed of sound assumed by the beamformer to construct phase delays. These two values should be the same, but the modeling capability to mimic errors in the data processing is provided.
- The center frequency and bandwidth of the signal. The beam patterns are sampled at 1 Hz intervals across the bandwidth of a broadband signal and averaged. The primary purpose is to approximate the effect of a broadband signal that brackets the design frequency of an array.

The beam pattern $\mathcal{B}(\psi, \psi_s)$ for a uniformly spaced linear array is given by

$$\mathcal{B}(\psi, \psi_s) = |b(\psi, \psi_s)|^2 , \tag{17}$$

where ψ_s is the steering angle, and b is given by

$$b(\psi, \psi_s) = \sum_{m=1}^{N} w_m \exp\left(i(m-1)d2\pi f \left[\frac{\sin\psi}{c} - \frac{\sin\psi_s}{c_0} \right] \right) . \tag{18}$$

In Eq. (18),

N = number of elements in array,

w_m = weight applied to m^{th} element for shading (i.e., Uniform, Hamming, Hanning) and/or normalization,

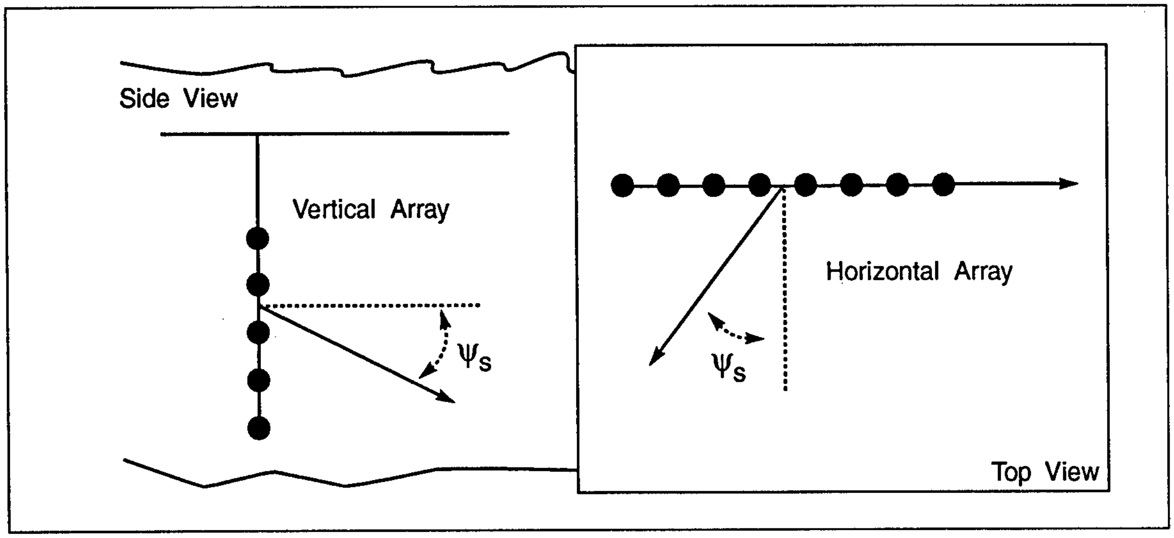


Fig. 15 – Conventions for specifying the positive steering angle ψ_s for beam patterns for horizontal and vertical linear arrays. For each array, the dotted line corresponds to broadside. For the vertical array, ψ_s is measured positive down. For the horizontal array, ψ_s is measured positive aft, and thus depends on the heading for the array. The specification of the steering angle does not depend upon the tilt of the array.

d = array spacing (between elements (m)),
c = actual speed of sound at array depth (m/s),
c_0 = assumed speed of sound at array depth (m/s) used to construct phase delays (usually $c_0 = c$, see discussion below), (19)
f = acoustic frequency (Hz),
ψ_s = steering angle, direction of main response axis of beam pattern measured with respect to array axis, $-90° \le \psi_s \le 90°$, and
ψ = angle with respect to array axis (for rays, map the vertical arrival and azimuthal angles into ψ), $-90° \le \psi \le 90°$.

Equation (18) corresponds to a plane wave beamformer. It is assumed that a plane wave of unit magnitude is incident on the array at an angle ψ, and phase delays are added to account for the wave reaching each element at slightly different times. (For details, see Ref. 4, p. 55.)

The beam patterns for the source and receiver arrays are not normalized in the same way. Receiver beam patterns, regardless of the type of spatial shading and number of dead elements, are normalized so that the main response axis is unity (i.e., 0 dB). It is assumed that the consequences of shading and dead elements are compensated for via the gain of the amplifiers or appropriate calibration factors. Source beam patterns are normalized according to the following conventions:

- Source beam patterns for uniform spatial shading and no dead elements are normalized so that the main response axis is unity (i.e., 0 dB).
- Source beam patterns for nonuniform spatial shading and no dead elements will include the reduction in level at the main response axis due to the shading (e.g., for Hanning, approximately −6 dB).
- Source beam patterns for nonuniform spatial shading and dead elements will include the reduction in level at the main response axis due to the shading (e.g., for Hanning approximately −6 dB) and the reduction due to the dead elements.

With these conventions, the total source level used for final scaling of reverberation and target echo should be the maximum possible for uniformly-shaded arrays and no dead elements since the calculated source beam pattern will compensate for the effects of spatial shading and dead elements. This assumes that the source level per element is the same for each element (i.e., variations are handled by the weights). Also source transmission loss levels that include the beam pattern can be directly compared to see the effect of the different configurations.

As mentioned in Section 4.5.1, the application of beam patterns in BIREV interacts with the angle used for the angular distribution of reverberation. When the reverberation time series for multiple receiver beams are to be calculated, only the source beam pattern is applied in BIREV. Received reverberation is distributed by mapping the vertical angle and the azimuthal angle of each ray arriving at the receiver line array into the angle between the ray and the line array. This array-raypath angle is then used by the program POSTBIREV to determine the appropriate beam weights for all of the receiver's beams. If both the source and receiver beam patterns are applied in BIREV (or when appropriate, TLGRID), then the reverberation time series may be distributed with respect to some "nontraditional" angle, such as vertical launch/arrival angles or angles related to the scattering process.

4.5.3 Scattering Strengths

The program will determine from the input TLGRID files the type of reverberation to be calculated and prompt the user to select the specific surface, bottom, or volume scattering model(s) to be used, including a bistatic component, when appropriate. Backscattering strengths may be specified as a constant scattering strength (in dB), in tabular form via a file containing a table of average grazing angle vs scattering strength, through a user-supplied subroutine (see Section A7), or derived from internally stored models.

<u>Bottom Scattering:</u> The primary bottom scattering model provided combines Lambert's law with a Kirchhoff's facet approximation for bistatic scattering [7]:

$$\sigma(\theta_{in}, \theta_{out}, \Psi) = \mu \sin\theta_{in} \sin\theta_{out} + \nu(1+\Delta\Omega)^2 \exp\left(-\frac{\Delta\Omega}{2\sigma^2}\right), \tag{20}$$

where θ_{in} and θ_{out} are the vertical incident and scattered grazing angles, respectively, and Ψ is the bistatic angle. The geometric factor $\Delta\Omega$ is given by

$$\Delta\Omega = \left(\cos^2\theta_{in} + cos^2\theta_{out} - 2\cos\theta_{in}\cos\theta_{out}\cos\Psi\right)/(\sin\theta_{in} + \sin\theta_{out})^2. \tag{21}$$

The first term in the sum of Eq. (20) corresponds to Lambert's law, and the second is the Kirchhoff facet approximation. The scattering parameters μ, ν, and σ are usually obtained from experimental data although for MacKenzie scattering, 10 $\log_{10}\mu = -27$ dB is used [3]. Guiding values for ν, and the angular spread factor σ are 10 $\log_{10}\nu = -10$ dB, and $\sigma \times 180/\pi = 10°$ [7].

Also provided for bottom scattering are four empirical bottom-type models from Urick [4] that correspond to bottom compositions of rock, sand, silt, and clay. These models use the average of the incident and the backscattered angle, with no bistatic dependence, and the internal tables are interpolated as required. The Kirchhoff facet approximation for bistatic scattering can be included if desired.

Surface Scattering: The internal surface scattering models provided are the Chapman-Harris empirical formula [2] and the Ogden-Erskine surface backscattering strength model [6]. Both require the wind speed in knots to be specified. The Chapman-Harris formula is provided for historical reasons, having been superceded by the Ogden-Erskine model. Both models use the average (arithmetic mean) of the vertical incident and the backscattered angle. Although neither the Chapman-Harris or the Ogden-Erskine model provide justification, the Kirchhoff facet approximation for bistatic scattering of Eq. (20) can be included if desired. Otherwise, no bistatic dependence is assumed.

Volume Scattering: For volume scattering, the user must provide a constant (column) scattering strength in dB for each individual scattering layer.

In general, spatial-dependence of the scattering strengths is handled in BIREV in a regional fashion by specifying the coordinates of polygons and the scattering model to be used when a scattering element is contained in that polygon. The coordinates for each polygon must be specified in a counterclockwise fashion, and one polygon should not contain another. Also, the user must specify a default scattering model to be used if a polygon cannot be found that contains the scattering element. This allows the user to tile the entire scattering region with polygons, or specify a single region as having a different scattering model. If the single bearing (2D) calculation approach is used, then range-dependence is accomplished by specifying the minimum and maximum ranges for each scattering model.

4.5.4 Optimization of Area Integration

The evaluation of Eqs. (5) or (7) is computationally intensive due to both the integration over scattering elements, and the summation over all possible multipaths from the source to the scatterer and then to the receiver. In general, the integration over scattering elements must be performed, since it is difficult to know a priori which elements can be ignored. The multipath summation though can spend considerable time evaluating contributions that are relatively weak and contribute minimally to the total reverberation. The optimization algorithm takes advantage of the fact that each ray path's propagation loss to a scatterer is known prior to integration to isolate those paths that contribute an insignificant amount to the reverberation. Using propagation loss and approximate scattering strength, the contributions from a scatterer are sorted from most significant to least significant. The smallest contributions can then be neglected according to the level of accuracy desired. With this method, the user may trade accuracy for computation time when necessary. The remainder of this section is adapted from Ref. 17.

We initially assume that dispersion due to the different travel times can be ignored and that the scattering function is approximately separable, i.e.,

$$\sigma(\theta_{in}, \theta_{out}) \approx f(\sin\theta_{in})g(\sin\theta_{out}). \qquad (22)$$

For convenience, we shall also assume omnidirectional beam patterns for the source and receiver, although this is not necessary.

For an arbitrary scattering patch, where there are N rays incident on the patch and M rays scattered from the patch to the receiver, the contribution to reverberation from the pairing of individual ray paths can be separated into components corresponding to the i^{th} incident and j^{th} scattered rays:

$$R_{ij} = F_i H_j \,, \tag{23}$$

where

$$F_i = f(\sin\theta_i)/L_i \,, \tag{24a}$$
$$H_j = g(\sin\theta_j)/\tilde{L}_j \,, \tag{24b}$$

and L_i and $\tilde{L}_j$ are the transmission losses from the source to the scattering patch and the patch to the receiver, respectively.

Let the sets $\{F_i\}$ and $\{H_j\}$ each be sorted into nonincreasing sequences, $\{\widetilde{F}_i : 1 \le i \le N\}$ and $\{\widetilde{H}_j : 1 \le j \le M\}$. Then a reordered set of contributions, $\{\widetilde{R}_{ij} = \widetilde{F}_i\widetilde{H}_j : 1 \le i \le N, 1 \le j \le M\}$, can be constructed that satisfy the following inequalities for all i and j:

$$\begin{aligned} \widetilde{R}_{i,j} &\ge \widetilde{R}_{i,j+1} \,, \\ \widetilde{R}_{i,j} &\ge \widetilde{R}_{i+1,j} \,. \end{aligned} \tag{25}$$

If T is the exact total contribution to reverberation from the patch, then $T = \sum_{i=1}^{N}\sum_{j=1}^{M}\widetilde{R}_{ij}$. Suppose $\widetilde{R}_{ij}$ has been calculated for the first n incident rays and m scattered rays, where $n < N$ and $m < M$. Let the calculated partial contribution to reverberation from the patch be denoted T_{nm}. Then $T_{nm} = \sum_{i=1}^{n}\sum_{j=1}^{m}\widetilde{R}_{ij}$ and T may be rewritten as:

$$T = T_{nm} + \sum_{i=1}^{n}\sum_{j=m+1}^{M}\widetilde{R}_{ij} + \sum_{i=n+1}^{N}\sum_{j=1}^{m}\widetilde{R}_{ij} + \sum_{i=n+1}^{N}\sum_{j=m+1}^{M}\widetilde{R}_{ij} \,. \tag{26}$$

Replacing each of the double summations above with an upper bound based on the inequalities of Eqs. (25), Eq. (26) can be rewritten as

$$T \le T_{nm} + U \tag{27}$$

where U is the upper bound on the contributions from the remaining $(N-n)$ incident and $(M-m)$ scattered rays given by

$$U = (M-m)\sum_{i=1}^{n}\widetilde{R}_{im} + (N-n)\sum_{j=1}^{m}\widetilde{R}_{nj} + (N-n)\,(M-m)\,\widetilde{R}_{nm} \,. \tag{28}$$

Note that U depends only on values of $\widetilde{R}_{ij}$ that have already been calculated for the current value of T_{nm}.

To estimate when the remaining contributions are insignificant, we define $E = 1 - T_{nm}/T$ as the relative error between the partial contribution T_{nm} from the patch and the actual contribution T from the patch. Similarly, we define $\overline{E} = 1 - T_{nm}/T_{nm} + U$ as the approximate relative error between the partial calculated contribution from the patch and the upper bound of the contribution from the patch. It can be shown that $E \le \overline{E}$. For a given patch, n and m are increased until the condition $\overline{E} \le \epsilon$ is met, where ϵ is a user-specified value less that one. The difference in dB between T_{nm} and T can be determined from the relationship $E = 1 - T_{nm}/T \le \overline{E} \le \epsilon$:

$$10\,\log_{10}\left(\frac{T}{T_{nm}}\right) \le 10\,\log_{10}\left(\frac{1}{1-\epsilon}\right) . \tag{29}$$

For $\epsilon = (0.1, 0.25, 0.5)$, Eq. (29) yields 0.46, 1.25, and 3.01 dB, respectively. Our current experience suggests that values of $\epsilon < 0.2$ are appropriate for environments that are slowly varying.

There are two limitations of this method. The first limitation is that contributions to reverberation are distributed in time whereas the above algorithm is applied at fixed ranges. Therefore neglected contributions from a fixed range may be significant (i.e., one of only a few) contributions for a particular time. This limitation becomes more important for pulses of short duration.

The second limitation is that the scattering function is assumed to be separable, whereas most scattering functions are not. This limitation is partially addressed as follows: we initially use Eq. (22) *only* when sorting the sets of incident and scattered components $\{F_i\}$ and $\{H_j\}$ but then use the *complete* scattering function to calculate the contributions $\{\widetilde{R}_{ij}\}$. Still, when the separability requirement breaks down, the ordering of $\{\widetilde{R}_{ij}\}$ may be incorrect. As a result, the approximate error may not steadily decrease as more multipaths are incorporated into T_{nm}, thereby causing the calculation to terminate prematurely. For the scattering strength functions provided in BIREV, premature termination of a patch contribution to the reverberation is most likely to occur for bistatic source/receiver geometries when the Kirchhoff facet approximation in Eq. (20) is not negligible.

The primary impact of these limitations is on the accuracy of the finer details of a model prediction. Major returns and general trends in the reverberation are rarely affected for values of ϵ in the range 0.1 to 0.2. Thus, use of the algorithm should be evaluated against the purpose of the model prediction. The savings in computation time may be significant, especially if multiple executions for the same environment are required.

4.5.5 Target Echos

BIREV provides two methods for calculating target echoes as alternatives to the calculation performed in program ACTENV. For both, the program provides a function call for calculating the target strength for which the user must provide the actual model. Currently, a default function always returns a target strength of 0 dB.

The first method for calculating target echoes is for a target at a fixed location. The program evaluates Eq. (8) and calculates the received time series. The options for beam patterns and distribution angles are the same as for a reverberation calculation. The source and receiver input files from TLGRID must contain at least one radial each. If multiple radials are contained in the input files, BIREV will use the radials whose bearings are closest to the true bearings of the target relative to the source and receiver. The program also allows for a list of locations to be processed, but will still output only one file. If the locations are not sufficiently separated in range, the individual target returns may overlap in the output beam-time series.

The second method, used for general wide-area assessment of target echoes, proceeds as though there was a target at each of the scattering patches used to calculate reverberation. Since these returns would overlap in time if simply combined, the return from each target location is reduced to single values of level, angle, and time. The user has various options. For the level value, the user selects either: (1) the maximum (peak) level of the temporal envelope, (2) the average of the return, or (3) the echo spreading loss (peak level divided by the energy of the total return). For the value of the angle, the choices are: (1) the return angle relative to the receiver line array for the peak (array-ray path angle), (2) the average array-raypath angle of the return, or (3) the azimuthal angle or horizontal bearing of the target location relative to the receiver location. Similarly, the choices for the value of the time are: (1) the time of the peak, (2) the average time, or (3) a pseudo two-way travel time calculated from the source-to-target-to-receiver range and an average sound speed.

Once the results for a target location have been determined (i.e., the level, angle, and time), overlap in angle-time bin is still possible. Thus, two different quantities are calculated for each angle-time bin, the peak and the average over the levels. It is unlikely that an angle-time bin will have only one target return placed in it. Thus, comparing the peak with the average gives an estimate of the extent of area that detection is likely. That is, if the peak is much greater than the average, then only a few locations produced a strong return. If required, the counts used to calculate the average can be also be output.

Other analysis products are available depending upon the calculations of the level, angle, and time. For example, if the third option for both time and angle are selected, then the level will be stored in an angle-time bin that corresponds to its location with respect to the receiver. Whereas, if the time of the peak is used, then the effect of the dispersion due to the multipaths is considered. There is no provision for applying the receiver beam patterns in the general assessment method. In this, each target return is assumed to arrive on the main beam axis of the normalized receiver beam pattern.

4.5.6 Output Files

Program BIREV generates several output files. These include:

- An annotated log file of the responses to the program prompts that will prove useful if you need to run the program again with the same input.
- A print file containing a complete record of the program execution.
- Various reverberation or target echo files for input to programs POSTBIREV, REVPLOTS, or ACTENV. (See Section A7 on page 66 for more details.)

4.6 Program TLVSR

The purpose of TLVSR is to compute the transmission loss from the source (receiver) to a target/layer depth or boundary as a function of range from the source (receiver). Here, transmission loss is defined as the total loss suffered by the multiple ray paths that propagate from the source (receiver) to the target/layer depth or boundary. The input files are those produced by TLGRID. Transmission loss vs range may be computed for a single bearing or for the entire TLGRID file. The single-bearing data files of transmission loss from the source-to-target and (by using reciprocity) from the target-to-receiver can be used in ACTENV to evaluate Eq. (9) for target returns as a function of range for monostatic and quasi-monostatic geometries. If the entire TLGRID file is processed and mapped into plan view, then a wide-area assessment of target returns can be calculated using the sonar equation representation (see Eq. (34) in Section 4.7). Additionally, TLVSR will compute and print the ray arrival structure as a function of range.

Program TLVSR computes both coherent and incoherent transmission loss. Although both are plotted, only incoherent transmission loss is written to the output file for subsequent target-echo calculations using ACTENV. Incoherent transmission loss L_{incoh} at the target depth at range ρ is given by

$$\frac{1}{L_{\text{incoh}}(\rho)} = \sum_{i \in m(\rho)} \frac{b(\theta_i)}{L_i(\rho)} , \tag{30}$$

where $m(\rho)$ indexes all the ray paths at the target/layer depth and range location. Similarly, coherent transmission loss $L_{\text{coh}}(\rho)$ is given by

$$\frac{1}{L_{\text{coh}}(\rho)} = \left[\sum_{i \in m(\rho)} \left(\frac{b(\theta_i)}{L_i(\rho)} \right)^{1/2} \cos \Psi_i \right]^2 + \left[\sum_{i \in m(\rho)} \left(\frac{b(\theta_i)}{L_i(\rho)} \right)^{1/2} \sin \Psi_i \right]^2 , \tag{31}$$

where Ψ_i is the phase angle of the i^{th} path.

For a single bearing, the possible output from program TLVSR consists of:

- a data file of incoherent transmission loss as a function of range for input to ACTENV,
- a plot of incoherent and coherent transmission loss as a function of range (e.g., Fig. 21 on page 43), and
- a print file of ray arrival structure as a function of range.

If the entire TLGRID file is processed (i.e., all bearings), the possible output from program TLVSR consists of:

- a data file of incoherent or coherent transmission loss as a function of range and bearing suitable for plotting in a waterfall format, and
- a data file of incoherent or coherent transmission loss as a function of location (i.e., range and bearing have been mapped into a plan view). The wide-area assessment of target returns can be constructed from these files using utility programs to combine and scale the results.

4.7 Program ACTENV

Program ACTENV predicts the performance of a monostatic, quasi-monostatic, or bistatic active sonar system by assembling, scaling, and plotting the results of other programs of the BiRASP model. Its primary output is a plot that overlays the individual mean received-power vs time-after-transmission envelopes of surface reverberation, bottom reverberation, ambient noise, and target echoes on a common level-vs-time grid. Not all of these envelopes need to be processed and plotted, except ambient noise. For example, ACTENV is capable of plotting only surface reverberation (and ambient noise).

Reverberation envelopes are constructed from the results of BIREV or POSTBIREV. The target-echo envelope is constructed from the sonar equation representation of Eq. (8), where vertical-beam-weighted transmission losses have been computed by TLVSR. Target echoes are initially computed as a function of range and then converted to a function of time-after-transmission by using the approximation of 1480 m/s. It is also possible to plot the target-echo time series as calculated by BIREV or POSTBIREV. Ambient noise is assumed to be omnidirectional and time-independent and is determined from a user-specified constant.

Program ACTENV allows for a variety of signal types although prior to ACTENV, all calculations assume a finite-duration (gated) continuous wave (CW) pulse. Also, an omnidirectional or vertical source array is assumed, whereas the receiving array may be omnidirectional or horizontal. In the latter case, horizontal directivity is accounted for by applying a horizontal-receiving directivity index. (Also a reverberation envelope averaging option can be used to simulate the receiving beam pattern.) A constant value of target strength is used, although BiRASP could easily be modified to account for more complex target scattering functions. Results are presented in (quasi-) absolute rather than relative units of measure. Acoustic quantities are calculated in units of total mean-acoustic-power in the processing band after beamforming and matched, or FFT filtering. Furthermore, these quantities are normalized by the power of a plane wave, having an rms amplitude of one micropascal (μPa) and expressed in decibels. It is important to note that quantities are not expressed on a per-Hertz basis nor in terms of energy (which results from time integration).

Waveform Types and Processing Gains: Reverberation and target-echo envelopes are scaled according to one of three generic pulse types: gated-CW, impulsive, or frequency-modulation (FM) slide. In each case, a value for the transmit source level SL is required. Here, SL is defined as the total transmitted acoustic intensity on the main-beam axis of the source beam pattern (not on a per-Hertz basis), which is referenced to a unit distance of 1 m from the acoustic center of the source. Program ACTENV requests the input of a source level SL_e of a single (and common) element and the number N of elements in the vertical source array, then computes SL from the relationship

$$SL = SL_e + 20 \ \log_{10} N \ . \tag{32}$$

Also, an analysis bandwidth W is input. For CW and FM pulses, W is also taken to be the signal bandwidth. For a CW pulse, W is actually calculated from an input signal duration. W is also used to scale an input omnidirectional noise level (per Hertz) to the analysis band.

The BiRASP model is primarily designed for the case of a finite-duration CW pulse. In ACTENV, the pulse duration, or length, D is input, and the matched signal and analysis bandwidths are found from $W = 1/D$.

Although traditionally an energy calculation, the BiRASP model approximates the performance of an impulsive source in terms of power by using a short CW pulse (in program BIREV) having a duration of $D = 1/f_b$, where f_b is the fundamental bubble-pulse frequency. Typically, D will range between 5 and 40 ms. For the impulsive signal case, ACTENV requests the value of the analysis bandwidth W. The strength of an impulsive source is usually available in units of energy-flux-density S_e. The user is advised to derive and input an equivalent-CW source level from the relationship

$$SL = 10 \ \log_{10} (S_e \ W/D) \ . \tag{33}$$

The FM signal option is intended to account for the matched filtering of a controlled waveform of duration D and bandwidth W. Calculations by BIREV should be made by using an equivalent-CW pulse of duration $D_e = 1/W$. Program ACTENV internally increases SL by 10 $\log_{10}(WD)$, which is the intended effect of pulse compression and is the source of the WD processing gain of a matched filter.

Wideband controlled pulses require a relatively short, integration range-step in program BIREV (see Eq.(16) with $D = D_e = 1/W$) and, consequently, can result in a significantly increased computation time. As an alternative, one can use the actual pulse duration D in BIREV and the actual source level SL in ACTENV, then specify the signal as being an impulsive type in ACTENV. The resulting reverberation envelope will require less computation time (because of larger range-integration step-size) and will be approximately the same level but will exhibit less fine structure than the compressed-pulse result because of the range-averaging effect of the longer pulse length). Unfortunately, target echoes will not be increased by the $10\log_{10}(WD)$ processing gain. However, this can be rectified by artificially increasing the target strength.

Spatial Scaling: When the BiRASP model is executed in the "Single Bearing Mode," it is designed primarily for an omnidirectional or vertical source array and for a omnidirectional or horizontal receiver. The vertical directivity of the source is accounted for in program TLGRID or BIREV. Program ACTENV accounts for any horizontal directivity of the receiver by applying an input (horizontal) RDI to received reverberation and ambient noise. An additional way of accounting for the horizontal directivity of reverberation is to use the option of performing a weighted-average of several reverberation envelopes. For example, envelopes calculated along different bearings (e.g., main and sidelobes) of the receiver beam pattern can be appropriately weighted by the beam pattern response and summed to approximate the horizontal directivity of the receiver. This is particularly useful in situations of potential sidelobe contamination. Other purposes of averaging reverberation envelopes include accounting for distinct reverberation returns in the ambiguous main beams of a horizontal receiver and averaging over several bearings within a relatively wide main beam.

Temporal Averaging: It is often desirable to include the smoothing effects of averaging received reverberation over the analysis time. POSTBIREV provides this capability, however, temporal averaging cannot be performed in ACTENV. For "Single Bearing Calculations," temporal averaging can be accomplished in program BIREV. The averaging is performed over the interval Δt between the equally spaced times at which reverberation is calculated. Therefore, it is particularly recommended that the interval Δt in BIREV be set to the analysis time in the case of an impulsive source.

Target-Echo Returns: ACTENV provides an additional method to those in program BIREV for calculating a target-echo return envelope. Using transmission losses computed by program TLVSR, ACTENV computes echo returns, ER, from the simple sonar equation

$$ER = SL - TLS + TS - TLR + PG\,, \tag{34}$$

where SL is total source level, TS is (a constant) target strength, TLS and TLR are transmission losses from source-to-target and target-to-receiver, respectively, PG is processing gain (if any), and all quantities are expressed in decibels. SL (actually found from SL_e and N), TS, and (indirectly) PG are user-supplied constants. Note that two (source and receiver) transmission loss files are required. The exception is the monostatic case (with identical source and receiver vertical beam patterns) where a single transmission loss file may be used.

Reverberation is calculated as a function of time (after transmission). Transmission loss is calculated as a function of range. To put echo returns on a common basis with reverberation, a source-to-target-to-receiver range is approximated at the discrete times for which received reverberation is calculated. Then the appropriate transmission-loss vs range files are interpolated to the respective component ranges, and an echo level is computed from Eq. (8) and plotted.

Upon completion, ACTENV will output

- a plot of surface, bottom, and volume reverberation, ambient noise, and target echo (e.g., Fig. 22 on page 44, Note: requires DISSPLA), and
- a print file of data detailing the time series for the surface, bottom, and volume reverberation envelopes and for the target echo.

4.8 Program POSTBIREV

The primary purpose of POSTBIREV is to apply the beam patterns to the reverberation and target echo angle-time series produced by BIREV. POSTBIREV is also used to perform various manipulations of the files containing the time series, such as calculating the total volume reverberation by summing the contributions from each layer, applying a time-averaging window, mimicking a ping repetition rate, shifting the origin of the time axis, and constructing multistatic results. It can also be used to extract single beams or bearings for input to ACTENV. The program REVPLOTS is used to scale the output time series from POSTBIREV to absolute levels and plot them in a waterfall or plan-view format.

The program is a set of subroutines driven by a menu. This enables the user to string a sequence of operations together. Most of the subroutines require only a few inputs, i.e., file names and one or two numbers. The exception is the subroutine to calculate the beam pattern tables, but this subroutine usually is executed only once. The input files to POSTBIREV are the raw (uncalibrated) time series calculated by BIREV or its own output files. The remainder of this section gives the name of the subroutine and a brief summary of its purpose.

- CREATE_BP_DATABASE_FILE

  Creates a look-up table of multiple beam patterns for a specific array. The number of beam patterns can be based on steering angles or beams equally spaced in "cosine space." Each beam pattern is sampled at a 0.5° resolution. This resolution is generally sufficient for arrays of up to 128 elements. (See Section 4.5.2 for details on the beam patterns supported.)

- CREATE_BP_PRINT_FILE

  Calculates a specific beam pattern and writes it to a print file for inspection.

- BEAMFORM_RAW_BIREV_DATAFILE

  Applies all the beam patterns contained in a beam pattern database file to the reverberation data contained in a raw BIREV datafile. Sums reverberation over launch/arrival angles for each time increment, thus reducing the data to reverberation vs time vs beam (or steering angle).

- SUM_RAW_BIREV_DATAFILES

  Takes two or more raw BIREV datafiles, possibly of different temporal lengths, adds them together, and outputs the result in raw BIREV datafile format.

- SUM_BEAMFORMED_BIREV_DATAFILES

  Takes two or more beamformed BIREV datafiles, possibly of different temporal lengths, adds them together, and outputs the result in beamformed BIREV datafile format.

- COLLAPSE_RAW_BIREV_FILE

  When no beam pattern is applied in BIREV, its output has generally been distributed over an "array-raypath" angle measured between a ray and the source or receiver array. This subroutine sums the raw BIREV data over its angular distribution and outputs a single time series as though the source or receiver array was omnidirectional. Its output can be plotted by ACTENV.

- RESTRICT_ANALYSIS_TIME_WINDOW

  Extracts a portion of the time series from each beam in a beamformed BIREV datafile.

- APPLY_TIME_AVERAGING_WINDOW

  Applies a forward-looking, user-specified time-averaging window (s) to a beamformed BIREV datafile. Outputs the results in beamformed BIREV datafile format.

- MIMIC_PING_REPETITION_RATE

  Mimics ping repetition rate by adding a portion of a beamformed BIREV datafile onto itself using a user-input ping repetition rate (s).

- SCALE_AND_SHIFT_BF_DATA

  Allows the user to scale the data in a beamformed BIREV datafile by a specified number of dB and shift it in time by a specified number of seconds. After scaling and shifting, writes out the result in beamformed BIREV datafile format.

- EXTRACT_BEAMFORMED_TIME_SERIES

  Extracts individual beam-time-series data from a beamformed BIREV datafile and writes them to a file which can be plotted using ACTENV.

4.9 Program REVPLOTS

REVPLOTS processes the reverberation and target echo beam-time series from POSTBIREV (or the angle-time series that would be input into POSTBIREV) by scaling them to absolute levels and plotting the results in a waterfall or plan-view format. Details on the scale factors are given in Section 4.7 except that those concerning spatial factors are not pertinent.

The plan-view format requires the user to specify the dimensions of a rectangular region in the appropriate coordinate system and the number of pixels in the x and y directions. The receiver-to-pixel bearing and the travel time from source-to-pixel-to-receiver are calculated and used to look up the information from the input files. The user has the option of specifying one or two input files for use in the plan view. The two-file option is used to resolve the left/right ambiguity of a horizontal line array.

The BiRASP distribution package also contains utility programs to manipulate these files (e.g., scale the level, threshold or add in a noise floor, add files, etc.).

4.10 Program ANGPLOTS

The program ANGPLOTS processes the reverberation and target-echo time series from BIREV that have been distributed by an angle other than the angle with respect to the source/receiver line array (e.g., a vertical arrival file). It performs the scaling to absolute levels and plots the results in a color waterfall format.

5. SAMPLE OUTPUT

This section presents a representative sample of the various plots produced by BiRASP. Two sample executions are presented. The first is for a "Single-Bearing Calculation" and is a repeat of the the example done in Ref. [1]. The second is for a bistatic source/receiver geometry. The source code distribution package for the BiRASP model will include the input files and logs of the program executions for these examples.

5.1 Single-Bearing Calculation Example

Surface and bottom reverberation and target echo will be calculated for a quasi-monostatic geometry. Table 3 lists the source, receiver, target and environmental parameters that are used. Seven sample plots are presented. The following brief descriptions are provided to orient the reader. Complete information is found in Section 4.

Figure 16 presents a composite environmental plot created by PROFIL. The sound-speed profiles are plotted on the same graph as the bathymetry with sound speed for zero depth located at the range of the profile. Additionally, the minimum sound speed and the bottom depth on the profiles are connected to one another. In this case, only two profiles were present.

Figure 17 is a plot of the raytraces (originating from the source) that are generated by RAYACT. The horizontal line at 100-m depth is the target depth. Only every 10^{th} ray was plotted.

Figures 18 through 20 are order contours generated in the RTHETA module. Figure 18 shows the source-to-surface contours; Fig. 19 shows the source-to-bottom contours; Fig. 20 shows the source-to-target contours. The source-to-surface and source-to-target plots indicate the presence of caustics where there are x's. In each case, the minimum ray order is 0, and the maximum ray order is 120. Additionally, each case processed all of the fan angles.

Figure 21 is a transmission loss vs range plot generated in the TLVSR module. It presents the transmission loss of the coherent and incoherent energy from the source to the target. Note that the transmission loss plotted includes the beam pattern response for the source as applied in program TLGRID.

Finally, the ACTENV module generates the reverberation plot shown in Fig. 22. This plot shows the surface and bottom reverberation and the target echo levels. These levels are presented as the total beampower at the output of the receiver and are a function of the time after transmission. Also shown on this plot is the ambient noise. Because ambient noise is independent of range, it is a straight line. Since the receiver is omnidirectional, neither the surface or bottom reverberation envelopes are weighted.

5.2 Bistatic Geometry Example

Surface, bottom, and volume reverberation and target echo are calculated for a bistatic source and receiver located approximately 200 km southeast of George's Bank. Three layer depths are used for calculating the volume reverberation. Also, multiple bottom backscattering regions are used. MacKenzie scattering is used for the majority of the area, but above the 3000-m contour

Table 3 – Parameters Used in the Quasi-Monostatic Sample Execution

SOURCE	
Frequency	: 500 Hz
Type	: vertical array of three elements
Steering	: 0° (broadside)
Element spacing	: $\lambda/2$ (λ = acoustic wavelength)
Element source level	: 200 dB
Depth	: 200 m (at center of array)
Pulse type	: CW
Pulse duration	: 5 s
RECEIVER	
Type	: horizontal (modeled as omnidirectional, corrected using Directivity Index)
Depth	: 250 m (at center of array)
Directivity Index	: 10 dB
TARGET	
Depth	: 100 m
Strength	: 15 dB//μPa
ENVIRONMENTAL	
Ambient noise	: 65 db/Hz//μPa (omnidirectional)
Range	: 200 km
Sound-speed profiles	: Range-dependent (see sample input file in Section A2, page 52)
Bottom bathymetry	: Linear upslope (see sample input file in Section A2, page 52)
Bottom loss (FNWC table)	: 3
Bottom backscatter	: Urick clay
Surface backscatter	: Chapman-Harris
Wind speed	: 20 knots

on the southern seamounts, scattering strengths for Urick rock are used. Table 4 lists the source, receiver, target, and environmental parameters used for the modeling. Figure 23 summarizes the results for the bistatic example. These results include:

- Total reverberation and the individual surface, bottom, and summed volume-layers contributions in plan view. Note: No beamforming has been done to these plots. The received angle with respect to the horizontal line array has been used as though it were the azimuthal angle, which makes the east-west orientation of the receiver array readily apparent.
- Wide-area assessment of the target echo through explicit multipath evaluation
- Total reverberation beam-time series in a color waterfall format.
- The bathymetry for the area as extracted from DBDB5 [9].

6. ACKNOWLEDGMENTS

This work was supported by the Office of Naval Research, Code 321. The authors are grateful to Dr. Fred T. Erskine of the Naval Research Laboratory for his continuous support and encouragement.

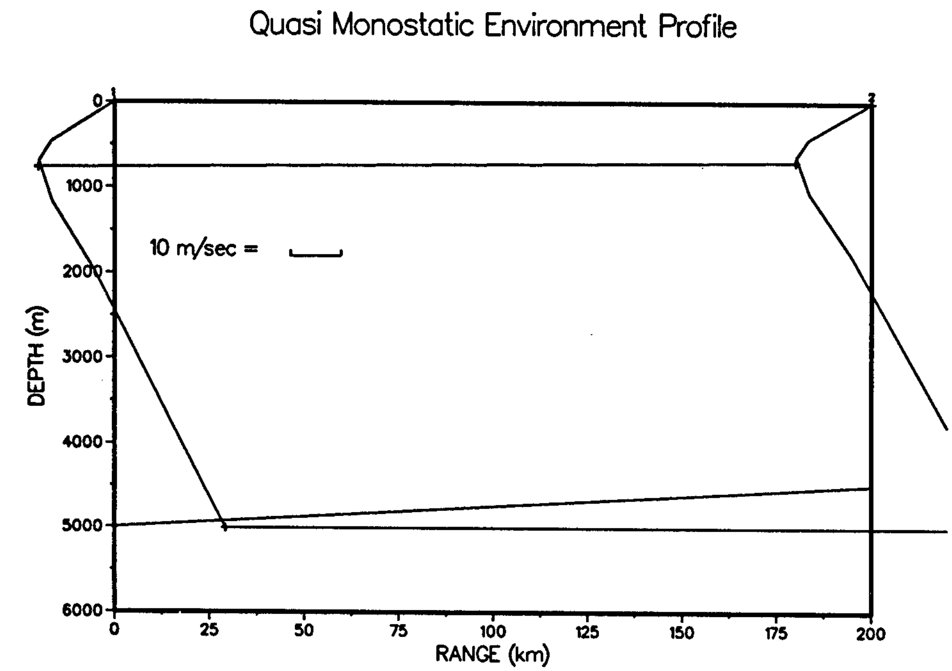


Fig. 16 – Composite environmental plot

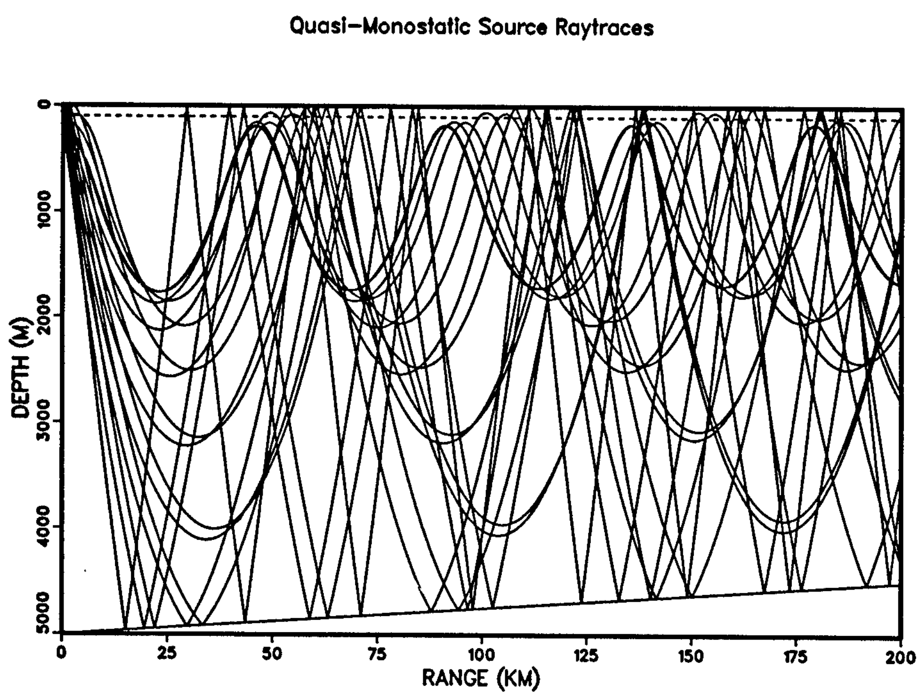


Fig. 17 – Raytraces from source. Dashed line indicates target depth.

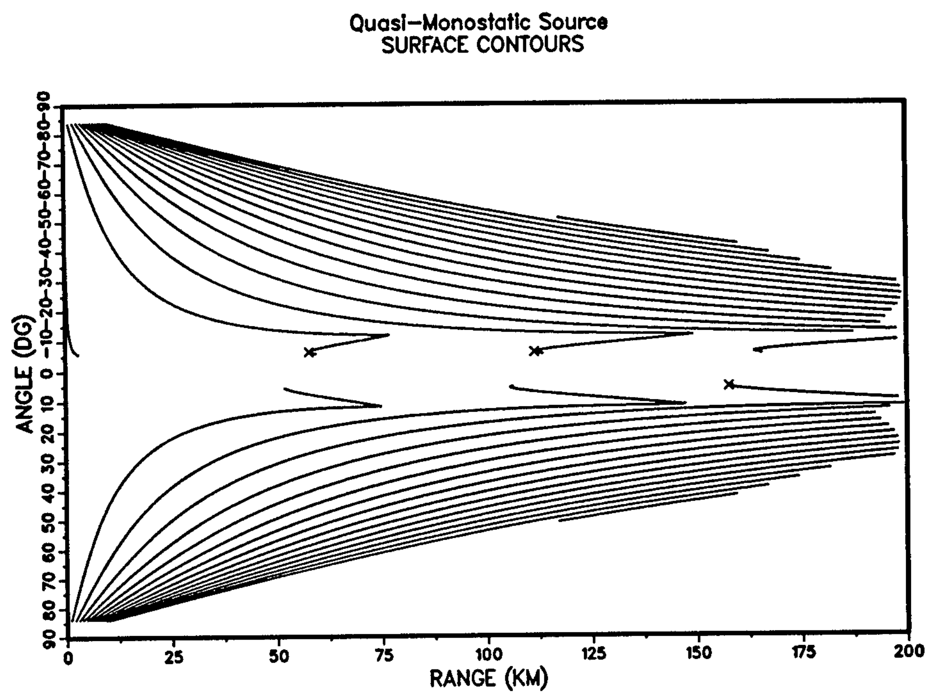


Fig. 18 – Source-to-surface ordered contours. Note: caustics are indicated using x's.

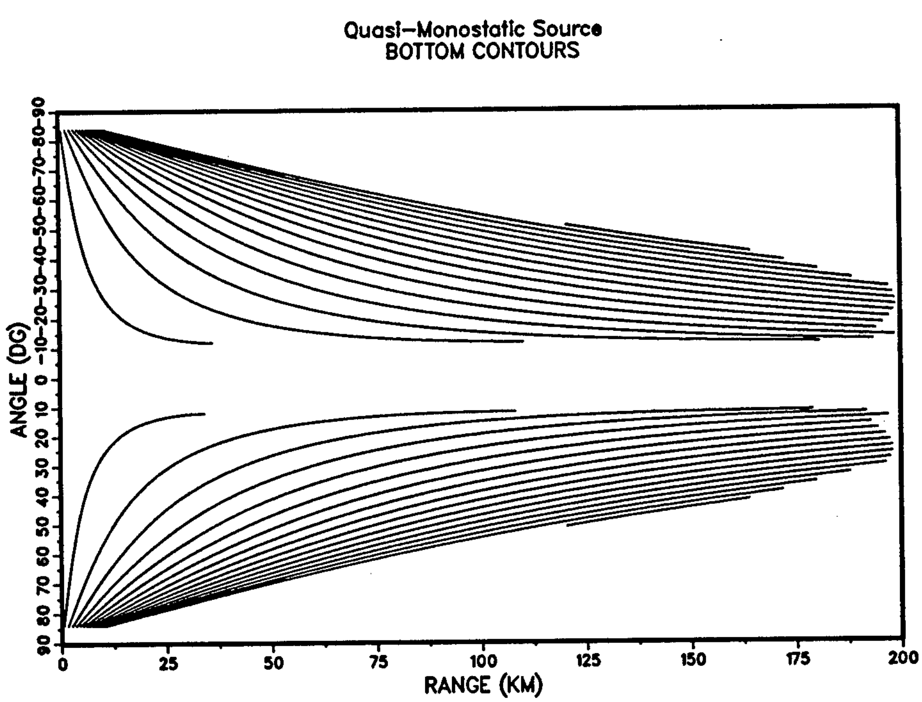


Fig. 19 – Source-to-bottom ordered contours

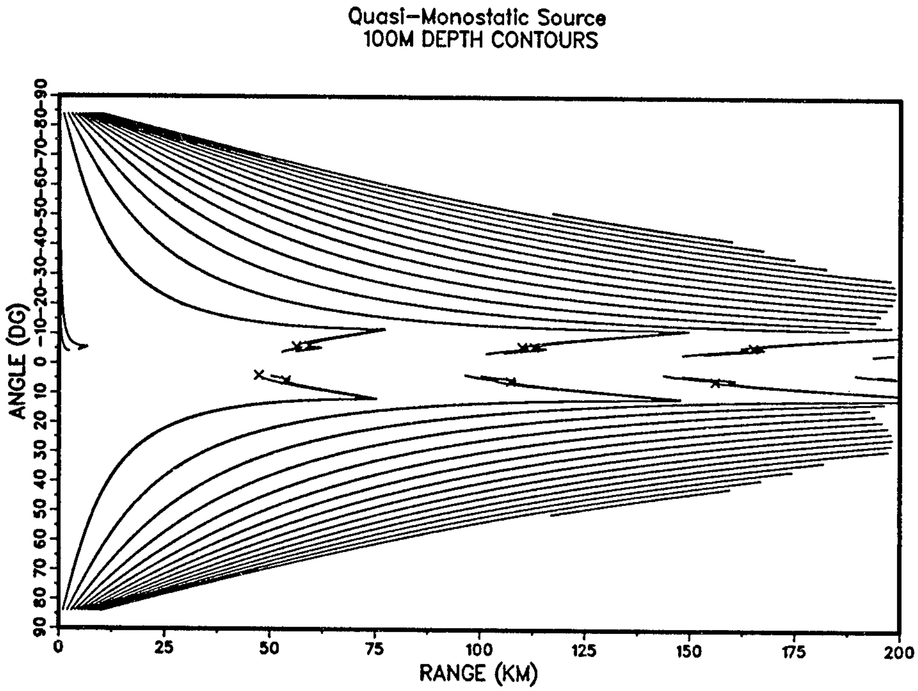


Fig. 20 – Source-to-target ordered contours

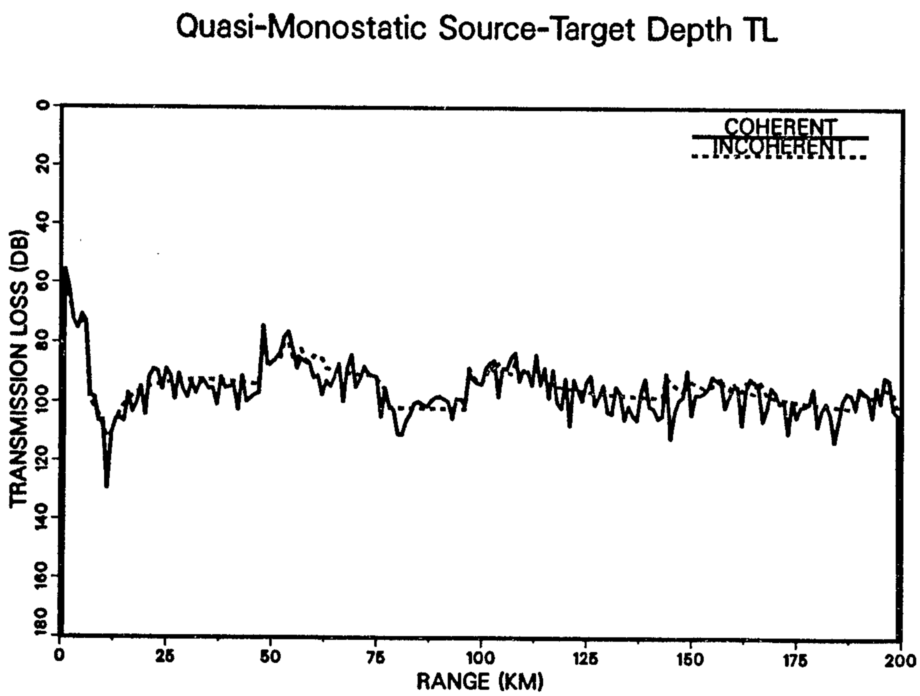


Fig. 21 – Transmission loss for source to target

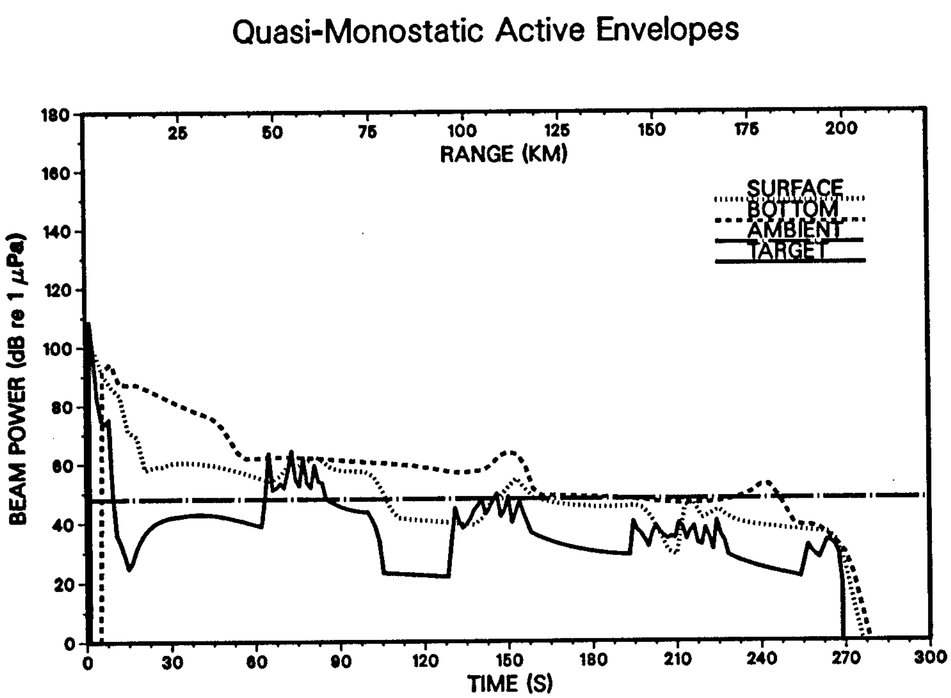


Fig. 22 – Surface and bottom reverberation levels; target echo levels vs the ambient noise

Table 4 – Parameters Used in the Bistatic Sample Execution

SOURCE	
Location	: 45° N 65° W, Heading 45° true
Frequency	: 200 Hz
Type	: Vertical array of 20 elements
Tilt from vertical	: 0.0°
Steering	: 11° up toward the ocean surface
Element spacing	: $\lambda/2$ (λ = acoustic wavelength)
Element source level	: 200 dB
Depth	: 200 m (at center of array)
Pulse type	: CW
Pulse duration	: 5 s
RECEIVER	
Location and Heading	: 39.75° N 64.75° W, heading due East
Type	: Horizontal array of 128 elements
Tilt from horizontal	: 0.0°
Element spacing	: $\lambda/2$ (λ = acoustic wavelength)
Depth	: 250 m (at center of array)
TARGET	
Depth	: 100 m
Strength	: 15 dB//μPa
ENVIRONMENTAL	
Ambient noise	: 65 db/Hz//μPa (omnidirectional)
Range	: 300 km ($\approx$ 400-s round trip time)
Sound-speed profiles	: Range-dependent – Summer profiles extracted from GDEM [8]
Bottom bathymetry	: Range-dependent – Extracted from DBDB5 [9]
Bottom loss (FNWC table)	: 3
Bottom backscatter	: MacKenzie except for Urick rock on southern seamounts
Surface backscatter	: Ogden-Erskine
Wind speed	: 10 knots
Volume backscatter	: Layers at 50, 100, and 150 m: Column Scattering Strength −65 dB

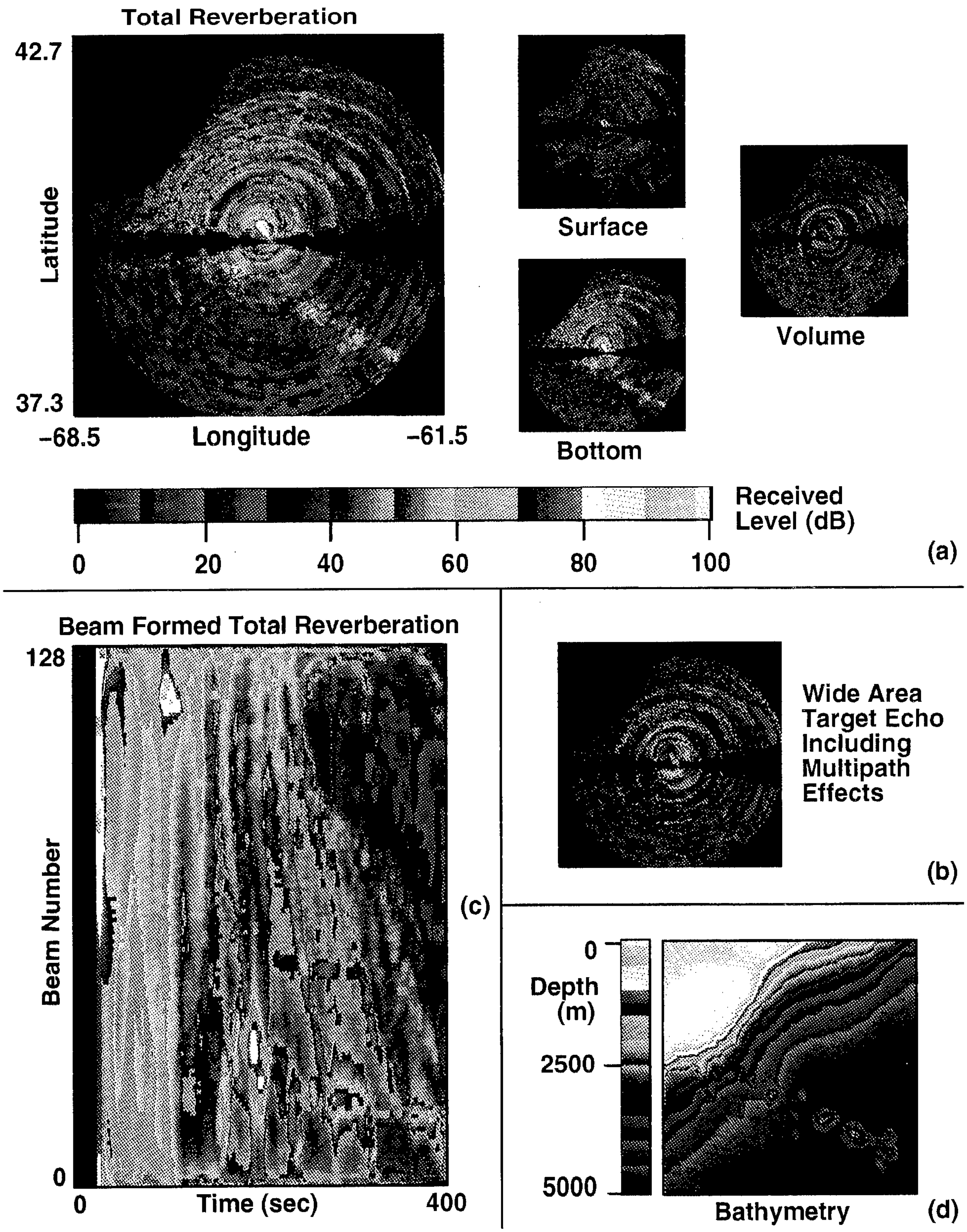


Fig. 23 – (a) Total reverberation in plan view (before beamforming). Also shown are the individual surface, bottom, and total-volume components. (b) Wide-area assessment of target echo that include shifts in apparent target location due to multipath effects. (c) Total reverberation as a function of beam and time. (d) The bathymetry for the region (from DBDB5).

REFERENCES

1. L.B. Palmer and D.M. Fromm, "The Range-Dependent Active System Performance Prediction Model (RASP)," NRL/FR/5160-92-9383, 21 July 1992.
2. R. P. Chapman and J. H. Harris, "Surface Backscattering Strengths Measured with Explosive Sound Sources," J. Acoust. Soc. Am. **34**, 1592–1597 (1962).
3. K.V. MacKenzie, "Bottom Reverberation for 530 and 1030-cps Sound in Deep Water," J. Acoust. Soc. Am. **33**, 1498–1504 (1961).
4. R.J. Urick, *Principles of Underwater Sound,* 3rd ed. (McGraw-Hill, N.Y., 1983), pp. 271–280.
5. "Acoustic Performance Prediction (APP) Historical Ocean Environmental Data Base Specification," Naval Ocean Research and Development Activity, Technical Report 137, 1982.
6. P.M. Ogden and F.T. Erskine, "An Empirical Algorithm for Low-Frequency Acoustic Surface Scattering Strengths," NRL/FR/5160-92-9377, 28 April 1992.
7. D.D. Ellis and D.R. Haller, "A Scattering Function for Bistatic Reverberation Calculations," J. Acoust. Soc. Am. **82**(S1), S124(A) (1987).
8. "Data Base Description For Master Generalized Digital Environmental Model (GDEM) Version 3.0," OAML-DBD-20D, Naval Oceanographic Office, Stennis Space Center, MS, Sept. 1990.
9. "Digital Bathymetric Database - 5.0 Minute Resolution (DBDB5)," Commanding Officer, U. S. Naval Oceanographic Office, Stennis Space Center, Mississippi.
10. C.W. Spofford, R.R. Greene, and J.B. Hersey, "The Estimation of Geo-Acoustic Ocean Sediment Parameters from Measured Bottom Loss Data," Technical Report SAI-83-879-WA, SAIC, McLean, VA, 1983.
11. E.R. Franchi, J.M. Griffin, and B.J. King, "NRL Reverberation Model: A Computer Program for the Prediction and Analysis of Medium-to-Long Range Boundary Reverberation." NRL Report 3721, May 1984.
12. J.J. Cornyn, "GRASS: A Digital Computer Ray Tracing and Transmission-Loss-Prediction System," NRL Formal Report 7621, December 1973.
13. H. Davis, H. Fleming, W.A. Hardy, R. Miningham, and S. Rosenbaum, "The Hudson Laboratories Ray Tracing Program," Hudson Laboratories, Columbia University, Technical Report 150, June 1968.
14. W.H. Thorp, "Deep Ocean Sound Attenuation in the Sub- and Low-Kilocycle-per-sec Region," J. Acoust. Soc. Am. **38**, 648–654 (1965).
15. C.W. Spofford, "The FACT Model, VOL I," Report 109, Ocean Science Program, Maury Center for Ocean Science, Department of the Navy, Washington, DC, November 1974.
16. L.B. Palmer, "Statistical Averages as Estimate for Acoustic Intensity," J. Acoust. Soc. Am. **53**, 300(A), (1973).
17. D.M. Fromm, K.H. Luther, and B.J. Orchard, "Computational Efficiency In A Bistatic Range-dependent Normal-Mode Acoustic Reverberation Model," J. Acoust. Soc. Am. **91**, 2343(A), (1992).

This page intentionally left blank.

Appendix

DESCRIPTION OF PROGRAM INPUT VARIABLES

This appendix provides a listing of the BiRASP programs' input variables, their definitions, and allowed values. They are given in the order they appear when a program is executed. As such, this resource provides a user-oriented perspective of the available processing paths. The first section of this appendix gives an overview of the format used to describe the inputs. Table A1 provides the contents for the other sections.

Table A1 – Contents for Appendix

A1 Explanation of Input Description Format

This section presents a quick overview of the format used to describe the inputs to the various BiRASP programs. The format uses indenting, white space, font changes, and line indexing to guide the user through the input structure. Three types of input lines are defined; lines that must be included, single lines whose inclusion depends upon a condition, and groups of lines whose inclusion depends upon a condition. Table A2 summarizes the format and indexing scheme for these input line types. The remainder of this section presents additional details within the context of an example.

Table A2 – Summary Of Input Line Formats And Indexing

Line Label Format	Indexing Scheme	Line Type
LINE 1:	Integers	Always included
Line A:	Letters	Single line whose inclusion depends upon a condition
Line B.1:	Letters.Integers	Group of lines whose inclusion depends upon a condition

LINE 1: (2 Entries) Format: free

A_REQUIRED_INPUT_LINE = Note that the number of entries for an input line is indicated within parentheses whenever it is greater than one. Also shown is the FORTRAN format of the line that the program expects. All of the numeric input in the BiRASP programs are free-field format.

VARIABLE_NAME_AS_FOUND_IN_PROGRAM_CODE = The complete variable name as found in the program code is provided to facilitate code modifications, etc. On the right side of the equals sign is descriptive information specifying purpose, range of values, units, etc. The user would only enter the desired value and not the variable name or equals sign.

Line A: Format: free

(Include if A_REQUIRED_INPUT_LINE >0.)

AN_OPTIONAL_INPUT_LINE = This is a single input line that is only included when the condition, given in italics as above, is met.

Important information is inserted between input lines and formatted as shown. Often it explains that the inclusion of the group of input lines that follows is dependent upon a condition.

LINE 2: Format: A80

SPECIFIC_INPUTS_VARIABLE = This is an example of a variable with limited input options. The lines that follow, within the left bracket, list the possibilities (to the left of the " := ") that the user would type as an entry for this input variable.

`ONE VALUE`	:=	Specific ASCII phrases to be input are shown in a capitalized `TYPEWRITER` font.
`ANOTHER VALUE`	:=	Many of the flags in BiRASP are ASCII phrases. This facilitates using a file name as an input option and provides for self-documentation of the input files.
`ONE MORE VALUE`	:=	To keep the amount of typing from significantly increasing, the user only needs to enter enough characters to be unique. Blanks or spaces and capitalization are ignored in this comparison.
A file name	:=	Thus, responses `ONEV`, `A`, and `OnEm` constitute the minimal number of characters needed to select one of the above. More characters can be entered (e.g., the entire phrase). For this example in which a file name is requested, any ASCII entry that fails to select one of the above options would be used as the file name.

Include Lines B.1–B.2 if
Line 2: SPECIFIC_INPUTS_VARIABLE = `ONE VALUE`**.**

Line B.1: Format: free

GROUP_VARIABLE_1 = This line is one of a group of input lines that is only included when the condition, shown above in a large bold font, is met. A given input line may have several such conditions that the reader can determine by working backwards. The overall condition on a group of input lines always precedes the first line of the group.

Line B.2: Format: free

(Include if LINE 1: A_REQUIRED_INPUT_LINE ≤0.)

GROUP_VARIABLE_2 = This input line is one of a conditional group, but it also has a secondary condition that is given in italics as above. Note that the line label is included with the parameter, i.e., LINE 1: A_REQUIRED_INPUT_LINE. This will be done whenever the referenced variable is not nearby or ambiguous. Although there may be additional conditional lines within the main groups, the level of line indexing will not be any deeper (e.g., B.2.1).

A2 Program PROFIL

PROFIL prepares sound-speed profiles and bottom bathymetry for use in subsequent programs. It is a program to order environmental data, correct for Earth's curvature, interpolate between data points, calculate first- and second-order derivatives of environmental sound-speed profiles (storing these data on file for later use in program RAYACT), and display the data in various plot formats.

PROFIL Input Variables:

LINE 1: Format: A80

PNTFIL = Flag indicating whether a print file is created.

A file name	:=	Name of print file to create.
`NO PRINT FILE`	:=	Do not create print file.

LINE 2: Format: A80

PRFILE = Name for output environmental data file for input to RAYACT.

LINE 3: Format: A80

BTHFIL = Name of the file containing the input bathymetric data. The data should be formatted as sequential pairs of range and depth values. *A maximum of 800 bottom points are allowed.*

```
  0.0,    5000.0
200.0,    4500.0
```

At left is the bathymetry file used in the sample calculation of Section 5.1. It consists of two pairs of range (km) and depth (m).

LINE 4: Format: A80

SSPFIL = Name of the file containing the input sound-speed profiles. The data should be formatted as sequential pairs of depth and sound-speed values (at each range). Each complete set is to be preceded by the range and number of depth values for that set. A range-independent case requires only one profile. *A maximum of 100 profiles, each with a maximum of 800 points, is allowed.*

```
0.0, 6
     0.0,    1495.0
   471.0,    1480.0
   706.0,    1477.0
  1177.0,    1480.0
  2000.0,    1490.0
  5000.0,    1520.0
200.0, 6
     0.0,    1495.0
   424.0,    1480.0
   635.0,    1477.0
  1059.0,    1480.0
  1800.0,    1490.0
  4500.0,    1520.0
```

At left is the sample sound-speed profile file used in the sample calculation of Section 5.1. It consists of profiles at ranges of 0 km and 200 km. Each profile contains six pairs of depth (m) and sound speed (m/s).

LINE 5: (3 Entries) Format: free

PLOT_FLAG(I), (I=1,3) = Flags for composite environment plot, sound-speed contours, and plots of individual profiles, respectively. One entry for each of the three options is needed, e.g., 0,0,0 to turn all of them off. These plots require DISSPLA.

- 1 := For each, create the plot.
- 0 := For each, do not create the plot.

Line A: Format: A80

(Include when a composite environment plot is requested.)

PLOT_FLAG(1) = Flag indicating the type of composite environment plot to create.

- `PRESENTATION FORMAT` := Create presentation type plot.
- `DIAGNOSTIC FORMAT` := Create diagnostic type plot.

Line B: (2 Entries) Format: free

(Include when a composite environment plot or sound-speed contour plot is requested.)

RAXINC = Range increment of range axis (km).

RAXMAX = Maximum range of range (km).

Line C: (2 Entries) Format: free

(Include when a composite environment plot or sound-speed contour plot is requested.)

DPAXINC = Depth increment of depth axis (m).

DPAXMAX = Maximum depth of depth axis (m).

Line D: (3 Entries) Format: free

(Include when plot(s) of selected profile(s) is/are chosen.)

SSP_START = Number of first profile to be plotted.

SSP_END = Number of last profile to be plotted.

SSP_INC = Increment of profiles to be plotted.

Line E: Format: 18A4

(Include when any plot is chosen.)

PLOT_TITLE(I), (I=1,17) = Title (up to 68 characters) to be used by plotting routines. (Formatted according to DISSPLA requirements.)

PROFIL **Output:**

- Environmental information in data file PRFILE to be used as input for program RAYACT.
- Print file PNTFIL containing: bottom profile table of range and bottom depth; sound-speed profile tables(s) of depth, speed, gradient, and curvature; and information on profile connections made.
- Optional (PLOTS):
  1. Sound-speed profiles (linear fit) plus bathymetry.
  2. Environmental sound-speed contours plus bathymetry. Note: Requires DISSPLA.
  3. Individual sound-speed profiles (spline fit).

A3 Program RAYACT

RAYACT is a ray-tracing program to calculate ray-path encounters with surface, bottom, and target/layer depth encounters associated with selected rays. Bottom-loss data are entered at this stage (to reduce computations for rays with high losses). The purpose is to calculate ray paths and relevant parameters (e.g., the travel time, turning points, grazing angles) from the source or receiver to the boundaries and target/layer depths. The information is stored in files for later use in program RTHETA. For quasi-monostatic or bistatic configurations, RAYACT must be run for both the source and the receiver. If the calculation of volume reverberation requires more than three layer depths, then additional executions are required since RAYACT can only process three depths at a time.

RAYACT Input Variables:

LINE 1: Format: A80

PNTFIL = Flag indicating whether print file is created.

- A file name := Name of print file to create.
- `NO PRINT FILE` := Do not create print file.

LINE 2: Format: A80

PRFILE = Name of the input data file created during the preceding PROFIL run.

LINE 3: Format: free

ARRAYDEPTH = Source/receiver array depth (m). Use the depth at the center of the array.

LINE 4: (3 Entries) Format: free

TGT_DEPTHS(I) = Target/layer depths (m), where I = (1,3).

Enter three target/layer depths; enter 0 for the unused target/layer depths.

Eg., Enter 40,0,0 for one target at 40 m depth; enter 0,0,0 for no layers.

Note: When a target/layer depth or the spacing between target/layer depths is comparable to 20 m, then the parameter MIN_ARC_INC on Line B should be redefined.

LINE 5: (2 Entries) Format: free

RINIT = Initial/minimum range within the environmental file PRFILE to locate the source/receiver and begin raytrace calculations (km). RINIT should be set to 0.0 except for possibly single bearing calculations.

RMAX = Maximum range for raytrace calculations (km).

LINE 6: Format: A80

CHANGEDEF = Flag indicating that the default ray tracing parameters are to be redefined.

- `CHANGE DEFAULTS` := Yes.
- `KEEP DEFAULTS` := No.

Line A: (5 Entries) Format: free

(Enter when redefining default ray tracing parameters.)

SSP_INTERP_FLAG = Determines use of profile connections. *Default = 0.*

- 1 := Do not use. Program will perform horizontal interpolation of sound speed.
- 0 := Use profile connections for sound-speed interpolation. Program will perform range interpolation of sound speed along line connections.

MAX_BTM_BOUNCES = Maximum number of bottom reflections (< 97) before ray termination. *Default = 96.*

MAX_TVL_TIME = Maximum travel time after which a ray will be terminated (s). *Default* $= (RMAX - RINIT)/1.3$.

MAX_BTM_LOSS = Maximum bottom loss that a ray is allowed to accumulate (dB). *Default = 175 dB.*

MAX_BOUNCES_AND_TURNS = Maximum number of boundary interactions plus turning points (< 199). *Default = 198.*

LINE 7: Format: A80

CHANGEDEF = Flag indicating that the default ray iteration parameters are to be redefined.

`CHANGE DEFAULTS` := Yes.
`KEEP DEFAULTS` := No.

Line B: (5 Entries) Format: free

(Enter when redefining default ray iteration parameters.)

MIN_ARC_INC = Minimum allowable arc length increment (m). *Default = 20.*

MAX_ARC_INC = Maximum allowable arc length increment (m). *Default = 1000.*

BDRY_HIT_DIST = Vertical distance from boundary or target depth within which a ray is considered as encountering that boundary or depth (m). *Default = 0.2.*

SSPACC = Minimum acceptable sound-speed field accuracy in interpolated values (m/s). *Default = 0.2.*

MAX_SIN_CHANGE = Maximum allowable change in $\sin(\theta)$ for rays of launch angle θ near 0.0° (wrt horizontal). *Default = 0.02.*

LINE 8: Format: A80

LAUNCH_ANGLE_FLAG = Flag indicating how launch angles are to be determined.

`AUTOMATIC` := Automatic ray selection by program.
`SHORT RANGE DEFAULT` := Use default for short range raytracing.
`LONG RANGE DEFAULT` := Use default for long range raytracing.
A file name := Name of file containing the angle fans. This user-specified file can contain any number of ray fans as long as the total number of rays does not exceed 500, and the minimum allowable angular sampling resolution is 0.1°.

```
 7
   -84.0,   -33.0,   3.0
   -30.0,   -23.0,   1.0
   -22.0,   -15.0,   0.5
   -14.0,    14.0,   0.2
    15.0,    22.0,   0.5
    23.0,    30.0,   1.0
    33.0,    84.0,   3.0
```

At left is a sample ray fan file. The first line is the number of fans. Each fan consists of start, stop, and step values in degrees. The edges of the fans should not overlap. The particular values in the example correspond to the those used by the program for the option `LONG RANGE DEFAULT`.

LINE 9: Format: A80

SFC_LOSS_FILE = Flag indicating that surface loss is to be included in the calculation.

`NO SURFACE LOSS` := Surface loss is set to 0 dB.
A file name := Name of file containing, as a function of range, surface loss (dB) vs incident grazing angle (°).

```
1
0.0
 0.0     1.00
 1.0     1.10
 2.0     1.50
 3.0     1.53
      .
      .
      .
88.0     1.20
89.0     1.20
90.0     1.20
```

At left is a sample surface-loss file. The first line indicates the number of tables to follow. Each subsequent table begins with the minimum range (km) for the table followed by 91 pairs of incident angle (°) and corresponding surface loss value (dB). The first table should start at range 0.0 km. There can be a maximum of 600 tables. The angles should step from 0.0° to 90.0° by 1.0°.

LINE 10: Format: A80

BLOSS_TBL_ORIGIN = Flag indicating source of bottom loss table.

- `FNWC`n := Use corresponding standard FNWC bottom type where ($n = 1, 2, 3, 4, 5$).
- A file name := Name of file containing, as a function of range, bottom loss (dB) vs incident grazing angle (°). This file's format is the same as the surface loss file described on LINE 9.

Line C: (2 Entries) Format: free

(Include if BLOSS_TBL_ORIGIN = FNWCn, n=(1,2,3,4,5).)

FRQNCY = Frequency (Hz) for calculation of bottom loss from internal tables.

RNG_OF_BLOSS_TBL(1) = Maximum range (km) to which this table of bottom loss values applies.

LINE 11: Format: A80

SFCFIL = Flag indicating that surface encounters are to be saved.

- A file name := Name of the file which will contain surface encounters for input to RTHETA.
- `NO SURFACE FILE` := Surface encounters will not be saved.

LINE 12: Format: A80

BTMFIL = Flag indicating that bottom encounters are to be saved.

- A file name := Name of the file which will contain bottom encounters for input to RTHETA.
- `NO BOTTOM FILE` := Bottom encounters will not be saved.

Line D: Format: A80

(Include when nonzero target/layer depths have been entered on LINE 4 and repeat once for each nonzero value.)

TGTFIL(j) = Flag indicating that target/layer depth encounters are to be saved.

- A file name := Name of the file which will contain the j^{th} target/layer depth encounters for input to RTHETA.
- `NO TARGET FILE` := Encounters will not be saved for this target/layer depth.

LINE 13: Format: free

STAT_FLAG = Flag indicating that the ray statistics (detailed ray boundary and target/layer depth encounter information) should be printed to the file "ray_stats.dat". *This can be a very large file and is generally produced only for diagnostic purposes.*

$$\begin{cases} 0 & := \text{No, do not create file.} \\ n > 0 & := \text{Print turning point and ray information for every } n^{\text{th}} \text{ ray.} \end{cases}$$

LINE 14: Format: free

PLOT_FLAG = Flag indicating whether a plot of ray paths will be created.

$$\begin{cases} 0 & := \text{No plot.} \\ n > 0 & := \text{Plot every } n^{\text{th}} \text{ ray.} \end{cases}$$

Input Lines E.1–E.3 when PLOT_FLAG $\neq$ 0.

Line E.1: (2 Entries) Format: free

MIN_LAUNCH_ANG = Minimum launch angle to be plotted (deg $> -90°$).

MAX_LAUNCH_ANG = Maximum launch angle to be plotted (deg $< 90°$).

Line E.2: (2 Entries) Format: free

MAX_RNG_TO_PLOT = Maximum range for depth axis on plot (km).

MAX_DP_TO_PLOT = Maximum depth for depth axis on plot (m).

Line E.3: Format: 18A4

PLOT_TITLE(I), (I=1,17) = Title (up to 59 characters) to be used by plotting routines. (Formatted according to DISSPLA requirements.)

RAYACT **Output:**

- Ray calculations in data files SFCFIL, BTMFIL, TGTFIL(j) for input to program RTHETA.
- Print file PNTFIL containing information for source or receiver: range and depth; fan angle specifications table; target depth(s); bottom loss table(s); brief summary of input environment data; ray path limits; iteration parameters; ray launch angles list.
- Optional output includes:
  1. Ray statistics printed to file "ray_stats.dat".
  2. Plot of ray paths in range, depth. Note: Requires DISSPLA.

A4 Program RTHETA

RTHETA is a program to reformat ray information from RAYACT, to determine the presence of caustics, and subsequently to apply the wave-theoretic correction (if requested). This program also calculates attenuation, and computes ray amplitudes and phases. Specifically, it forms range/source-angle (order contour) curves corresponding to either surface (z = 0) or bottom encounters or arrivals at predetermined depth. Transmission loss is calculated by using both ray-bundle (with caustic correction) and spatial averaging techniques. The $R(\theta, \rho)$ curves, together with one type of transmission loss estimate, are stored on an output file for further processing (by TLGRID).

RTHETA Input Variables:

LINE 1: Format: A80

PNTFIL = Flag indicating whether print file is created.

$\begin{cases} \text{A file name} := \text{Name of print file to create.} \\ \texttt{NO PRINT FILE} := \text{Do not create print file.} \end{cases}$

LINE 2: Format: A80

INFILE = Name of the input data file created by preceding RAYACT run. See LINES 11-D of RAYACT input description.

LINE 3: Format: A80

OUTFIL = Name for the data file which will contain the output $R(\theta, \rho)$ curve data for input to TLGRID.

LINE 4: Format: free

FRQNCY = Source (receiver) frequency (Hz). The frequency is used to calculate attenuation via Thorp's equation and in the caustic correction.

LINE 5: Format: A80

TL_TYPE = Type of transmission loss estimates stored on an output file OUTFIL.

$\begin{cases} \texttt{SPATIALLY AVERAGED TL} := \text{Normally used with bottom returns.} \\ \texttt{RAY BUNDLE TL} := \text{Normally used with surface and target/layer depth returns.} \end{cases}$

LINE 6: Format: A80

CHANGEDEF = Flag indicating whether to redefine the default RTHETA processing parameters.

$\begin{cases} \texttt{CHANGE DEFAULTS} := \text{Yes.} \\ \texttt{KEEP DEFAULTS} := \text{No.} \end{cases}$

Line A: (5 Entries) Format: free

(Enter when changing the default RTHETA *processing parameters.)*

MIN_ORDER = Minimum order of $R(\theta, \rho)$ curves to process. This is approximately the minimum number of turning points and boundary encounters (i.e., ray reversals of a ray path). *Default = Minimum = 1.*

MAX_ORDER = Maximum order of $R(\theta, \rho)$ curves to process. This is approximately the maximum number of turning points and boundary encounters (i.e., ray reversals of a ray path). *Default = Maximum = 120.*

ANGLE_FLAG = Flag indicating the launch angles to be processed. *Default = 0.*

$\begin{cases} 0 := \text{All available angles (as determined by program RAYACT) are processed.} \\ 1 := \text{Angular bounds will be input.} \end{cases}$

ANGS_TO_DELETE = Number of discrete launch angles to be deleted in construction of $R(\theta, \rho)$ curves. *Default = 0, Maximum = 20.*

CAUSTIC_FLAG = Flag indicating whether or not to include bottom caustics in output file OUTFIL. This flag only has effect when TL_TYPE = `RAY BUNDLE TL`. The default is No.

$\begin{cases} 1 := \text{Yes.} \\ 0 := \text{No.} \end{cases}$

Line B: (2 Entries) Format: free

(Include when ANGLE_FLAG > 0 .)

MIN_LAUNCH_ANG = Minimum launch angle to be processed (°).

MAX_LAUNCH_ANG = Maximum launch angle to be processed (°).

Line C: Format: free

(Include when ANGS_TO_DELETE > 0, and repeat ANGS_TO_DELETE times.)

ANGLE = Launch angle (°) to be deleted from $R(\theta, \rho)$ curves.

LINE 7: Format: A80

PLOT_FLAG = Flag indicating whether a plot of the order contours should be created.

$\begin{cases} \texttt{PLOT CONTOURS} := \text{Yes.} \\ \texttt{NO PLOTTING} := \text{No.} \end{cases}$

Input Lines D.1–D.3 when PLOT_FLAG = `PLOT CONTOURS`.

Line D.1: (3 Entries) Format: free

RMIN = Minimum range on plot axis (km).

RMAX = Maximum range on plot axis (km).

RINC = Range increment on plot axis (km).

Line D.2: (3 Entries) Format: free

ANGMIN = Minimum launch angle on plot axis (°).

ANGMAX = Maximum launch angle on plot axis (°).

ANGINC = Launch angle increment on plot axis (°).

Line D.3: Format: 18A4

PLOT_TITLE(I), (I=1,17) = Title (up to 68 characters) to be used by plotting routines. (Formatted according to DISSPLA requirements.)

RTHETA Output:

- Plot of ordered contours. Note: Requires DISSPLA.
- Calculations of $R(\theta, \rho)$ curves and associated transmission losses in data file OUTFIL for input to programs TLGRID.
- Print file PNTFIL containing: the input characteristics, launch angles processed, contour information, and ray histories and events listed by ray order and family type.

A5 UNIX Script RASPLOOP

In general, the calculation of Eqs. (5), (7), or (8) require that raytraces be performed for numerous bearings radiating from both the source and the receiver. The UNIX shell script RASPLOOP is designed to facilitate the multiple executions of the programs PROFIL, RAYACT, and RTHETA.

Inputs to RASPLOOP consist of sample template files containing the inputs that the user would normally provide to the actual executables. The only constraint on the contents of these files is that file names that should be differentiated by different bearings have the ASCII string "0000" in them. RASPLOOP will read bearings from a user-provided file and construct file names by substituting the actual bearing for the string "0000". The user can also input a list of files to delete after each bearing is completed. This allows intermediate files to be removed.

RASPLOOP Input Variables:

LINE 1: Format: A80

PROFIL_INPUT_FILE = Name of template file containing inputs to PROFIL.

```
NO PRINT FILE
PROFIL_SRC.TMP
B0000S.DAT
SVP_INSITU.DAT
0 0 0
```

At left is a sample PROFIL template file. The bathymetry file, B0000S.DAT, will cycle through the bearings but the sound-speed file, SVP_INSITU.DAT, will not.

LINE 2: Format: A80

RAYACT_INPUT_FILE = Name of template file containing inputs to RAYACT.

```
NO PRINT FILE
PROFIL_SRC.TMP
530.0
100.0  0.0  0.0
0.0  150.0
KEEP DEFAULTS
KEEP DEFAULTS
LONG RANGE DEFAULT
NO SURFACE LOSS
BL0000S.DAT
RAYACT_SRC_SUR.TMP
RAYACT_SRC_BOT.TMP
RAYACT_SRC_DEP100.TMP
0
0
```

At left is a sample RAYACT template file. The bottom-loss file, BL0000S.DAT, will cycle through the bearings.

LINE 3: Format: A80

RTHETA_SURFACE_INPUT_FILE = Name of template file containing inputs to RTHETA for processing the surface encounters recorded by RAYACT.

- NO := Indicates that NO surface files are to be processed.
- A file name := Indicates that surface files are to be processed.

```
NO PRINT FILE
RAYACT_SRC_SUR.TMP
RT0000.SRC.SUR
500
RAY BUNDLE TL
KEEP DEFAULTS
NO PLOTTING
```

At left is a sample RTHETA template file for processing surface encounters. The output file, `RT0000.SRC.SUR`, must cycle through the bearings since it needs to be saved for later input to TLGRID.

LINE 4: Format: A80

RTHETA_BOTTOM_INPUT_FILE = Name of template file containing inputs to RTHETA for processing the bottom encounters recorded by RAYACT.

- NO := Indicates that NO bottom files are to be processed.
- A file name := Indicates that bottom files are to be processed.

```
NO PRINT FILE
RAYACT_SRC_BOT.TMP
RT0000.SRC.BOT
500
SPATIALLY AVERAGED TL
KEEP DEFAULTS
NO PLOTTING
```

At left is a sample RTHETA template file for processing bottom encounters. The output file, `RT0000.SRC.BOT`, must cycle through the bearings since it needs to be saved for later input to TLGRID.

LINE 5: Format: A80

RTHETA_DEPTH1_INPUT_FILE = Name of template file containing inputs to RTHETA for processing the encounters recorded by RAYACT for first target/layer depth.

- NO := Indicates that NO target/layer files are to be processed for the first depth.
- A file name := Indicates that target/layer files are to be processed for the first depth.

```
NO PRINT FILE
RAYACT_SRC_DEP1.TMP
RT0000.SRC.DEP1
500
RAY BUNDLE TL
KEEP DEFAULTS
NO PLOTTING
```

At left is a sample RTHETA template file for processing target/layer depth encounters. The output file, `RT0000.SRC.DEP1`, must cycle through the bearings since it needs to be saved for later input to TLGRID.

LINE 6: Format: A80

RTHETA_DEPTH2_INPUT_FILE = Name of template file containing inputs to RTHETA for processing the encounters recorded by RAYACT for second target/layer depth.

- NO := Indicates that NO target/layer files are to be processed for the second depth.
- A file name := Indicates that target/layer files are to be processed for the second depth. See LINE 5 for a sample file.

LINE 7: Format: A80

RTHETA_DEPTH3_INPUT_FILE = Name of template file containing inputs to RTHETA for processing the encounters recorded by RAYACT for third target/layer depth.

$\begin{cases} \text{NO} & := \text{Indicates that NO target/layer files are to be processed for the third depth.} \\ \text{A file name} & := \text{Indicates that target/layer files are to be processed for the third depth. See LINE 5 for a sample file.} \end{cases}$

LINE 8: Format: A80

BEARING_INPUT_LIST_FILE = Name of file containing list of bearings to be processed.

```
0.0
45.0
90.0
135.0
180.0
```

At left is a sample file listing the bearings to be processed. There should be one bearing per line. The script will convert the bearing into the "####" format (e.g., 90.0 becomes 0900), thus the bearing resolution should be no higher than 0.1°.

LINE 9: Format: A80

FILE_DELETE_LIST = List of file names to be deleted after each bearing is completed. For the sample files given above, inputting "*.TMP" would remove the intermediate files created by PROFIL and RAYACT.

RASPLOOP Output:

In addition to the output generated by the executables, PROFIL, RAYACT, and RTHETA, RASPLOOP creates for each RTHETA process selected an ASCII file containing a list of the bearings and corresponding RTHETA output files. These files, which can be used as input to TLGRID, are named "??????.brg_list" where the question marks correspond to the names of the RTHETA template files.

A6 Program TLGRID

The program TLGRID processes the output of RTHETA. The typical use is to generate two sectored polar-coordinate systems, one each centered at the source position and at the position of the receiver. Typically from 72 to 360 radials are used, providing azimuthal sampling at 1°–5° intervals, with the density chosen depending upon the complexity of the environment. Sampling in range is typically performed at intervals of $\Delta\rho \leq c\tau/2$ where τ is the pulse duration. For each raypath to a given range bin, the acoustic intensity, travel time, grazing angle, and launch angle sampled in range and azimuth are stored in a lookup file for later use.

It is possible to apply beam patterns in TLGRID although this should only be done in special cases, such as for vertical arrays.

TLGRID also allows the "encounters" in each range bin to be sorted according to various criteria. For example, when the integration optimization algorithm in program BIREV is to be used, the contributions are sorted according to the acoustic intensity multiplied by the sine of the grazing angle, anticipating the probable magnitude of the contribution to the surface or bottom reverberation.

TLGRID Input Variables:

LINE 1: Format: A80

ARRAY = Flag indicating whether source or receiver RTHETA datafiles are being gridded.

$\begin{cases} \texttt{SOURCE} := \text{Source array.} \\ \texttt{RECEIVER} := \text{Receiver array.} \end{cases}$

LINE 2: (2 Entries) Format: free

ARRAYS_LATITUDE = The array's latitude, in decimal degrees, and should range between $-90°$ and $90°$, south and north, respectively.

ARRAYS_LONGITUDE = The array's longitude, in decimal degrees, and should range between $-180°$ and $180°$, east and west, respectively.

LINE 3: Format: free

NUMBER_OF_RADIALS = The number of RTHETA datafiles pertaining to this source/receiver array. *(Maximum of 360 radials.)*

Line A: Format: A80

(Input if NUMBER_OF_RADIALS > 1.)

RADIAL_FILE = The name of the file containing the radial's bearings and corresponding RTHETA files.

```
0.0
RT0000.SRC.BOT
45.0
RT0450.SRC.BOT
90.0
RT0900.SRC.BOT
135.0
RT1350.SRC.BOT
180.0
RT1800.SRC.BOT
```

At left is a sample of a RADIAL_FILE containing five pairs (by lines) of bearing and the corresponding RTHETA filename. This file will automatically be produced if the UNIX script RASPLOOP is used. The list of files does not have to be ordered by increasing bearing since the program will perform a sort. This allows additional bearings to increase local azimuthal resolution to easily be added later.

Input Lines B.1 – B.2 if NUMBER_OF_RADIALS = 1.

Line B.1: Format: free

RADIAL_BEARING = The bearing, in degrees, of the radial. This bearing is specified in degrees clockwise from true north, ranging from 0° to 360°.

Line B.2: Format: A80

INFILE = The name of the RTHETA datafile corresponding to the radial.

LINE 4: Format: free

DR = The range step (km) used to grid the source/receiver array's RTHETA datafiles.

LINE 5: Format: free

NUM_HITS_PER_GRID_RANGE_TO_KEEP = The maximum number of scattering boundary or target/layer depth encounters per grid range that the user wants written to the TLGRID output datafile *after* the input data have been sorted by estimated strength of contribution. *Maximum of 800 "encounters" per grid range.*

LINE 6: Format: A80

SORTING_KEY = Flag indicating whether the "encounter" data at each grid range of each radial should be sorted solely by transmission loss or by the product of transmission loss and the sine of a encounter's grazing angle. See discussion of the integration optimization algorithm in BIREV.

- `TL` := Sort by TL only. Recommended for constant scattering strength functions.
- `MODIFIED TL` := Sort by TL $\times \sin(\textit{grazing angle})$. Recommended for angle-dependent scattering strength functions.

LINE 7: Format: A80

APPLY = Flag indicating whether or not to apply a vertical beam pattern to the source/receiver array's input data. Only ONE beam pattern can be applied in TLGRID.

- `APPLY BEAM PATTERN` := Apply a beam pattern.
- `NO BEAM PATTERN` := Do not apply a beam pattern.

Line C: Format: A80

(Include if APPLY = `APPLY BEAM PATTERN`*.)*

WHEN = Flag indicating whether to sort the input data *before* or *after* the vertical beam pattern has been applied.

- `SORT BEFORE BP` := Sort before applying the vertical beam pattern.
- `SORT AFTER BP` := Sort after applying the vertical beam pattern.

Line D: Format: A80

(Include if APPLY = `NO BEAM PATTERN`*.)*

WEIGHT = Flag indicating whether or not to weight the input data during the sorting process by a vertical beam pattern *without permanently applying the pattern.*

- `USE BP IN SORT` := Include beam pattern weights when sorting input data.
- `NO BP IN SORT` := Do not include beam pattern weights when sorting input data.

Line E: (2 Entries) Format: free

(Include if APPLY = `APPLY BEAM PATTERN` *or WEIGHT =* `USE BP IN SORT`*.)*

SHIPS_HEADING = The heading of the source/receiver array. This bearing is specified in degrees clockwise from true north, ranging from 0° to 360°.

TILT = The tilt, in degrees, of the source/receiver array. For a vertical array, the tilt is measured from the vertical with a positive tilt angle indicating that deeper elements lag behind the shallower elements in the direction the array is being towed. Figure A1 shows the conventions for the arrays' tilts.

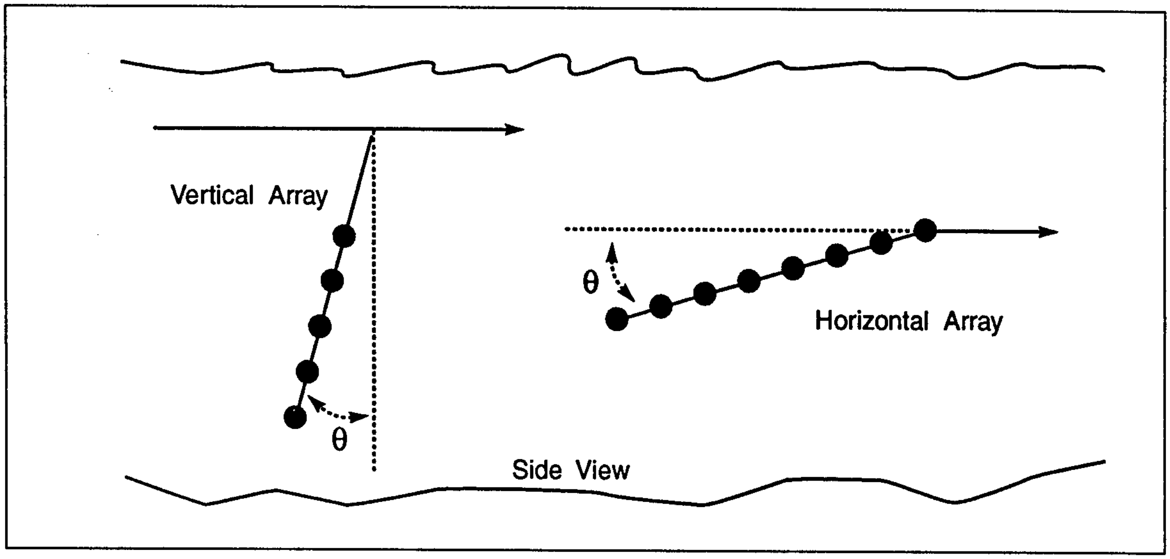


Fig. A1 – Conventions for specifying the positive tilt θ of horizontal and vertical linear arrays. Shown for each are the vertical plane containing the array, the solid horizontal lines that indicate the heading of the array, and the dotted reference lines for determining θ. An array's orientation specified using "pitch and roll" must be converted to this convention.

LINE 8: Format: A80

OUTFIL = The name of the TLGRID output datafile.

Include Lines F.1 – F.9 if APPLY = APPLY BEAM PATTERN **or WEIGHT =** USE BP IN SORT**, i.e., if applying a beam pattern, either permanently or just for sorting.**

Line F.1: Format: free

SIGNALS_CENTER_FREQ = The center frequency (Hz) of the signal transmitted by the source or received by the receiver.

Line F.2: Format: free

BANDWIDTH = The bandwidth (Hz) of the signal. This bandwidth is considered to be centered on SIGNALS_CENTER_FREQ. The beam patterns are sampled at 1 Hz intervals across the bandwidth of a broadband signal and averaged. The primary purpose is to approximate the effect of a broadband signal that brackets the design frequency of an array. No averaging is performed for bandwidths less than 1 Hz.

Line F.3: (2 Entries) Format: free

ACTUAL_SOUND_SPEED = The speed of sound (m/s) at the source/receiver array's depth.

ASSUMED_SOUND_SPEED = The speed of sound (m/s) used to calculate the phase factors for beamforming.

Note: In general, these will (and should) have the same value.

Line F.4: Format: free

NUMBER_OF_ELEMENTS(S_R_ARRAY) = The total number of source/receiver hydrophones.

Line F.5: Format: free

NUMBER_OF_BAD_ELEMENTS(S_R_ARRAY) = The number of inoperative hydrophones in the source/receiver array.

Line F.6: Format: free

(Include if NUMBER_OF_BAD_ELEMENTS(S_R_ARRAY) ≠ 0 and repeat NUMBER_OF_BAD_ELEMENTS(S_R_ARRAY) times.)

BAD_ELEMENTS(i) = The element number of the i^{th} inoperative source/receiver hydrophone.

Line F.7: Format: free

ELEMENT_SPACING(S_R_ARRAY) = The interelement spacing of the source or receiver's hydrophones in meters.

Line F.8: Format: A80

TYPE_OF_WEIGHTING(S_R_ARRAY) = Flag indicating the type of spatial shading to be applied to the source/receiver's beam pattern.

- `HAMMING` := Hamming weighting.
- `HANNING` := Hanning weighting.
- `UNIFORM` := Uniform weighting.
- A file name := Read in user-defined weights from a file. There should be NUMBER_OF_ELEMENTS(S_R_ARRAY) values contained in this file, one number per line. The first number corresponds to the spatial shading factor of the first hydrophone in the source/receiver array, etc.

Line F.9: Format: free

STEERING_ANGLE(S_R_ARRAY,1) = A source/receiver steering angle. For a vertical array, positive angles are directed down from broadside. For a horizontal array, positive angles are measured aft from broadside, and thus depends on the heading for the array. (For more details on conventions see Section 4.5.2 on page 26.)

TLGRID **Output:**

- Results datafile OUTFIL for input to program BIREV or TLPLOTS.

A7 Program BIREV

The program BIREV calculates raw (uncalibrated) reverberation or target echo levels as functions of time for arbitrary source/receiver geometries. A separate program execution is required for each scattering boundary or volume layer and for each target depth. Scattering strengths are calculated using the internally provided models or incorporated using the templates provided for user-supplied scattering-strength subroutines. Spatial dependence for the scattering strengths is handled in a regional fashion by specifying the coordinates of polygonal areas and the corresponding model to be used. The integration over scatters can span the full range and azimuth (centered on the receiver), or assuming azimuthal symmetry, span just the range from the receiver. Source and receiver beam patterns can be applied unless multiple beam results are desired. The program has many options and capabilities, and the user is strongly urged to review Section 4.5 on page 23.

The program stores results on output files for further processing (e.g., for the scaling and plotting performed by program POSTBIREV).

BIREV **Input Variables:**

LINE 1: Format: A80

LOGNAM = File name for an annotated log file of the responses to the program prompts which will prove useful if you need to run the program again with the same input.

LINE 2: Format: A

PFNAME = File name for a print file containing a complete record of the program execution.

LINE 3: Format: A

RRTNAM = Root or base name used to create the modeled reverberation's output file(s). See the output file description at end of the section.

LINE 4: Format: A

SFNAME = Name of output file from TLGRID containing information regarding the source.

LINE 5: Format: A

RFNAME = Name of output file from TLGRID containing information regarding the receiver.

LINE 6: Format: A80

COORDINATE_SYSTEM = Flag indicating whether planar/Cartesian or spherical/Great Circle distance calculations will be used.

- `CARTESIAN` := Cartesian coordinate system based on a Mercador projection about the receiver's location will be used.
- `SPHERICAL` := Spherical coordinate system will be used.

LINE 7: (3 Entries) Format: free

TMIN = The minimum time (s) after transmission for which returns are calculated.

TMAX = The maximum time (s) after transmission for which returns are calculated.

DT = The incremental time (s). For reverberation, recommend DT < SIGNALS_DURATION/2, and that DT > SIGNALS_DURATION acts as a time-averaging window. For a general wide-area assessment of target echo, DT should be set to approximately $\Delta r/(2c)$ where Δr is the range sampling rate (see LINE 4 of the inputs to TLGRID), and c is the speed of sound.

LINE 8: Format: free

SIGNALS_DURATION = The duration (s) of the transmitted pulse.

LINE 9: Format: A

RVB_OR_TE_FLAG = Flag indicating the type of calculation to be performed by the program.

- `REVERBERATION` := Calculate reverberation. Program determines type of reverberation from the input TLGRID files.
- `SINGLE TARGET ECHO` := Calculate target echo for a single target at a specific location.
- `GENERAL TARGET ECHO` := Calculate a general wide-area assessment of target echo.

Include Lines A.1 – A.3 if RVB_OR_TE_FLAG = `REVERBERATION`.

Line A.1: Format: A

(Include if the source/receiver geometry is monostatic or quasi-monostatic.)

INTEGRATION_TYPE = Flag indicating whether a general area or a single bearing integration is to be performed.

- `GENERAL AREA` := Perform the full integration over range and azimuth. This option is assumed if the source/receiver geometry is bistatic.
- `SINGLE BEARING` := Perform the integration over range only, assuming azimuthal symmetry.

Line A.2: Format: free

(Include if INTEGRATION_TYPE = `SINGLE BEARING`*.)*

SOURCE_BEARING = The bearing, in degrees, of the radial within the source file which the user wants to use for the single bearing integration.

This bearing is specified in degrees clockwise from true north, ranging from 0° to 360°.

Line A.3: Format: free

(Include if INTEGRATION_TYPE = `SINGLE BEARING`*.)*

RECEIVER_BEARING = The bearing, in degrees, of the radial within the receiver file which the user wants to use for the single bearing integration.

This bearing is specified in degrees clockwise from true north, ranging from 0° to 360°.

Include Lines B.1 – B.3 if RVB_OR_TE_FLAG = `GENERAL TARGET ECHO`.

Line B.1: Format: A

ECHO_DIST_FLAG = Flag indicating how each of the individual target echo envelopes are reduced to a single level value.

- 1 := Calculate the maximum (peak) level of the temporal envelope.
- 2 := Calculate the average of the return.
- 3 := Calculate the echo spreading loss (peak level divided by the energy of the total return).

Line B.2: Format: A

TIME_DIST_FLAG =.Flag indicating how each of the individual target echo envelopes are reduced to a single arrival time.

- 1 := Calculate the time of the peak of the temporal envelope.
- 2 := Calculate the average time over the temporal envelope.
- 3 := Calculate a pseudo two-way travel time calculated from the source-to-target-to-receiver range and an average sound speed.

Line B.3: Format: A

ANGLE_DIST_FLAG = Flag indicating how each of the individual target echo envelopes are reduced to a single received angle.

- 1 := Calculate the return angle relative to the receiver line array for the peak (array-ray path angle).
- 2 := Calculate the average array-raypath angle of the return.
- 3 := Calculate the azimuthal angle or horizontal bearing of the target location relative to the receiver location.

Line C: (2 Entries) Format: free

(Include if RVB_OR_TE_FLAG = `SINGLE TARGET ECHO`.*)*

TGT_Y = The latitude in decimal degrees of the target's location.

TGT_X = The longitude in decimal degrees of the target's location.

LINE 10: Format: A

ARRAYS_CONFIGURATION(SOURCE) = Flag indicating whether the source array is omnidirectional, horizontal, or vertical. If a source beam pattern was applied in program TLGRID, then `OMNIDIRECTIONAL` should be selected.

- `OMNIDIRECTIONAL` := Omnidirectional source.
- `HORIZONTAL` := Horizontal source.
- `VERTICAL` := Vertical source.

Input Lines D.1 – D.2 if ARRAYS_CONFIGURATION(SOURCE) ≠ OMNIDIRECTIONAL.

Line D.1: Format: free

SHIPS_HEADING(SOURCE) = The heading of the source array. This bearing is specified in degrees clockwise from true north, ranging from 0° to 360°.

Line D.2: Format: free

ARRAYS_TILT(SOURCE) = The tilt, in degrees, of the source array.

For a horizontal array, the tilt is measured from the horizontal, and a positive tilt angle indicates that aft elements are deeper than the fore elements in the direction the array is being towed, as is normally the case. For a vertical array, the tilt is measured from the vertical with a positive tilt angle indicating that deeper elements lag behind the shallower elements in the direction the array is being towed.

LINE 11: Format: free

ARRAYS_CONFIGURATION(RECEIVER) = Flag indicating whether the receiver array is omnidirectional, horizontal, or vertical. If a receiver beam pattern was applied in program TLGRID, then `OMNIDIRECTIONAL` should be selected.

`OMNIDIRECTIONAL`	:=	Omnidirectional receiver.
`HORIZONTAL`	:=	Horizontal receiver.
`VERTICAL`	:=	Vertical receiver.

Input Lines E.1 – E.2 if ARRAYS_CONFIGURATION(RECEIVER) ≠ `OMNIDIRECTIONAL`.

Line E.1: Format: free

SHIPS_HEADING(RECEIVER) = The heading of the receiver array. This bearing is specified in degrees clockwise from true north, ranging from 0° to 360°.

Line E.2: Format: free

ARRAYS_TILT(RECEIVER) = The tilt, in degrees, of the receiver array.

For a horizontal array, the tilt is measured from the horizontal, and a positive tilt angle indicates that aft elements are deeper than the fore elements in the direction the array is being towed, as is normally the case. For a vertical array, the tilt is measured from the vertical with a positive tilt angle indicating that deeper elements lag behind the shallower elements in the direction the array is being towed.

Line F: Format: A

Input Line F if Line 9:RVB_OR_TE_FLAG = `REVERBERATION` *or* `SINGLE TARGET ECHO`.

DISTRIBUTION_ANGLE = The angle by which the modeled reverberation or target echo will be distributed.

`ACTUAL ARRIVAL ANGLE`	:=	The actual angle with respect to the receive array. Receiver beam patterns will be applied using program POSTBIREV. Option not available for single bearing and omnidirectional receiver calculations.
`ACTUAL LAUNCH ANGLE`	:=	The actual angle with respect to the source array. Source beam patterns will be applied using program POSTBIREV. Option not available for single bearing and omnidirectional source calculations.
`VERTICAL LAUNCH ANGLE`	:=	The vertical launch angle from the source.
`VERTICAL ARRIVAL ANGLE`	:=	The vertical arrival angle into the receiver.
`GRAZING ANGLE`	:=	The incident grazing angle upon the scattering boundary.
`SCATTERED ANGLE`	:=	The reflected angle from the scattering boundary.
	:=	More options on next page.

`AVG BOUNDARY ANGLES`	:=	The average of the incident and reflected angles with respect to the scattering boundary.
`GEOMETRIC ANGLE`	:=	The arcsine of the geometric mean of the sines of the incident and reflected angles with respect to the scattering boundary.
`BISTATIC ANGLE`	:=	The bistatic angle. Not an option for monostatic and quasi-monostatic geometries.
`NO ANGLE`	:=	Calculate the reverberation as a function of time only. This option only available for single bearing calculations.

Input Lines G.1 – G.18 for source beam pattern inputs if ARRAYS_CONFIGURATION(SOURCE) $\neq$ `OMNIDIRECTIONAL` <u>and</u> DISTRIBUTION_ANGLE $\neq$ `ACTUAL LAUNCH ANGLE`.

Line G.1: Format: free

SIGNALS_CENTER_FREQ = The center frequency (Hz) of the signal transmitted by the source or received by the receiver.

Line G.2: Format: free

BANDWIDTH = The bandwidth (Hz) of the signal. This bandwidth is considered to be centered on SIGNALS_CENTER_FREQ. The beam patterns are sampled at 1 Hz intervals across the bandwidth of a broadband signal and averaged. The primary purpose is to approximate the effect of a broadband signal that brackets the design frequency of an array. No averaging is performed for bandwidths less than 1 Hz.

Line G.3: (2 Entries) Format: free

ACTUAL_SOUND_SPEED = The speed of sound (m/s) at the source/receiver array's depth.

ASSUMED_SOUND_SPEED = The speed of sound (m/s) used to calculate the phase factors for beamforming.

Note: In general, these will (and should) have the same value.

Line G.4: Format: free

NUMBER_OF_ELEMENTS(S_R_ARRAY) = The total number of source/receiver hydrophones.

Line G.5: Format: free

NUMBER_OF_BAD_ELEMENTS(S_R_ARRAY) = The number of inoperative hydrophones in the source/receiver array.

Line G.6: Format: free

(Include if NUMBER_OF_BAD_ELEMENTS(S_R_ARRAY) $\neq$ 0 and repeat NUMBER_OF_BAD_ELEMENTS(S_R_ARRAY) times.)

BAD_ELEMENTS(i) = The element number of the i^{th} inoperative source/receiver hydrophone.

Line G.7: Format: free

ELEMENT_SPACING(S_R_ARRAY) = The interelement spacing of the source or receiver's hydrophones in meters.

Line G.8: Format: A80

TYPE_OF_WEIGHTING(S_R_ARRAY) = Flag indicating the type of spatial shading to be applied to the source/receiver's beam pattern.

- `HAMMING` := Hamming weighting.
- `HANNING` := Hanning weighting.
- `UNIFORM` := Uniform weighting.
- A file name := Read in user-defined weights from a file. There should be NUMBER_OF_ELEMENTS(S_R_ARRAY) values contained in this file, one number per line. The first number corresponds to the spatial shading factor of the first hydrophone in the source/receiver array, etc.

Line G.9: Format: A

BEARS_OR_BMS(S_R_ARRAY) = Flag indicating whether the source/receiver's beam pattern(s) main response axis/axes will be specified by steering angle(s) or beam number(s).

- `STEERING ANGLES` := Specify by steering angle(s).
- `BEAMS` := Specify by beam number(s).

Include lines G.10 – G.13 if BEARS_OR_BMS(S_R_ARRAY) = `STEERING ANGLES`.

Line G.10: Format: A

SEQ_OR_SPEC = Flag indicating whether the number of source/receiver beam patterns will be determined by stepping through a sequence of steering angles or by specifying each steering angle.

- `SEQUENCE` := Step along a sequence of steering angles.
- `SPECIFY EACH` := Specify each steering angle.

Steering angle conventions are as follows: for a vertical line array, 0° is perpendicular to the array, 90° points directly downward, and −90° points directly upward; for a horizontal line array, 0° is broadside, 90° points directly aft of ship, and −90° points directly forward of ship.

Line G.11: (3 Entries) Format: free

(Include if SEQ_OR_SPEC = `SEQUENCE`*.)*

BEGSTER = The beginning steering angle, in degrees, used to denote the steering angle(s) of the source/receiver's beam pattern(s).

ENDSTER = The end steering angle, in degrees, used to denote the steering angle(s) of the source/receiver's beam pattern(s).

STERSTEP = The steering angle step, in degrees.

Input Lines G.12 – G.13 if SEQ_OR_SPEC = `SPECIFY EACH`.

Line G.12: Format: free

NUM_BEAM_PATTERNS(S_R_ARRAY) = The number of steering angles to be used to construct the source/receiver's beam pattern(s).

Line G.13: Format: free

(Repeat this line NUM_BEAM_PATTERNS(S_R_ARRAY) times.)

STEERING_ANGLE(SOURCE,i) = A source/receiver steering angle. For a vertical array, positive angles are directed down from broadside. For a horizontal array, positive angles are measured aft from broadside, and thus depends on the heading for the array. For more details on conventions, see Section 4.5.2 on page 26.

Include lines G.14 – G.18 if BEARS_OR_BMS(S_R_ARRAY) = `BEAMS`.

Line G.14: Format: A

BEAM_1_DIRECTION(S_R_ARRAY) = Flag indicating whether beam 1 of the source/receiver array is oriented forward (toward the tow-ship) or aft.

- `FORWARD` := Beam 1 forward.
- `AFT` := Beam 1 aft.

Line G.15: Format: A

SEQ_OR_SPEC = Flag indicating whether the number of source/receiver beam patterns will be determined by stepping through a sequence of beams or by specifying each beam.

- `SEQUENCE` := Step along a sequence of beams.
- `SPECIFY EACH` := Specify each beam.

Line G.16: (3 Entries) Format: free

(Include if SEQ_OR_SPEC = `SEQUENCE`.)

BEGBMS = The beginning beam number used to denote the beams of the source/receiver's beam pattern(s).

ENDBMS = The end beam number used to denote the beams of the source/receiver's beam pattern(s).

BMSTEP = The beam number step used to denote the beams of the source/receiver's beam pattern(s).

Input Lines G.17 – G.18 if SEQ_OR_SPEC = `SPECIFY EACH`.

Line G.17: Format: free

NUM_BEAM_PATTERNS(S_R_ARRAY) = The number of beams to be used to construct the source/receiver's beam pattern(s).

Line G.18: Format: free

(Repeat this line NUM_BEAM_PATTERNS(S_R_ARRAY) times.)

BEAMS(SOURCE,#) = A source/receiver beam number.

Input Lines G.1 – G.18 for receiver beam pattern inputs if ARRAYS_CONFIGURATION(RECEIVER) $\neq$ `OMNIDIRECTIONAL` and DISTRIBUTION_ANGLE $\neq$ `ACTUAL ARRIVAL ANGLE`.

Include Lines H.1 – H.4 if Line A.1:INTEGRATION_TYPE = `GENERAL AREA`.

Line H.1: Format: A

SECTORING_METHOD(SOURCE) = Flag indicating the means by which each radial contained in the source's input file is assigned an angular region (in degrees, directed outward from the source's position) within which that radial's information is deemed "correct".

- `AUTOMATIC SECTORING` := The program will determine the sector sizes of each radial.
- `MANUAL SECTORING` := User-input radial sector sizes. Program will perform error checking to avoid overlapping sectors, but will allow gaps.

Line H.2: (2 Entries) Format: free

(Include if SECTORING_METHOD(SOURCE) = `MANUAL SECTORING` *and repeat this line for as many radials are in the source's input file.)*

RADIAL_SECTOR_ANGULAR_RANGE(SOURCE,n,1) = The start bearing of the n^{th} source radial's sector.

RADIAL_SECTOR_ANGULAR_RANGE(SOURCE,n,2) = The end bearing of the n^{th} source radial's sector.

These bearings are specified in degrees clockwise from true north, ranging from 0° to 360°.

Line H.3: Format: free

SECTORING_METHOD(RECEIVER) = Flag indicating the means by which each radial contained in the receiver's input file is assigned an angular region (in degrees, directed outward from the receiver's position) within which that radial's information is deemed "correct".

- `AUTOMATIC SECTORING` := The program will determine the sector sizes of each radial.
- `MANUAL SECTORING` := User-input radial sector sizes. Program will perform error checking to avoid overlapping sectors, but will allow gaps.

Line H.4: (2 Entries) Format: free

(Include if SECTORING_METHOD(RECEIVER) = `MANUAL SECTORING` *and repeat this line for as many radials are in the receiver's input file.)*

RADIAL_SECTOR_ANGULAR_RANGE(RECEIVER,n,1) = The start bearing of the n^{th} receiver radial's sector.

RADIAL_SECTOR_ANGULAR_RANGE(RECEIVER,n,2) = The end bearing of the n^{th} receiver radial's sector.

These bearings are specified in degrees clockwise from true north, ranging from 0° to 360°.

Include Lines I.1 – I.14 if
Line 9:RVB_OR_TE_FLAG = `REVERBERATION`.

Line I.1: Format: free

CHOICE_OF_NUM_SCAT_PROCESSES = Flag indicating whether a single scattering model or multiple scattering models are to be used for the reverberation calculation.

`SINGLE SCATTERING PROCESS` := Use one scattering model.
`MULTIPLE SCATTERING PROCESSES` := Use multiple scattering models.

If CHOICE_OF_NUM_SCAT_PROCESSES = `SINGLE SCATTERING PROCESS` then process Lines I.5 – I.14

If CHOICE_OF_NUM_SCAT_PROCESSES = `MULTIPLE SCATTERING PROCESSES`, then process Lines I.2 – I.14.

Line I.2: Format: free

(Include if CHOICE_OF_NUM_SCAT_PROCESSES = `MULTIPLE SCATTERING PROCESSES`.*)*

NUM_SCATTERING_AREAS = The number of scattering regions into which the region of integration will be divided. Each of these scattering regions can have a different scattering model. These regions will be polygons for Line A.1:INTEGRATION_TYPE = `GENERAL AREA` and range brackets for Line A.1:INTEGRATION_TYPE = `SINGLE BEARING`. Full coverage of the region of integration is not necessary since an additional "default" scattering model will be requested.

Repeat Lines I.3 – I.14 NUM_SCATTERING_AREAS times.

Line I.3: (2 Entries) Format: free

(Include if Line A.1:INTEGRATION_TYPE = `SINGLE BEARING`.*)*

MIN_DIST_SCATTERING_PROCESS = The minimum range (km) measured radially outward from the receiver, delimiting the beginning of a scattering region.

MAX_DIST_SCATTERING_PROCESS = The maximum range (km) measured radially outward from the receiver, delimiting the ending of a scattering region.

Line I.4: (2 Entries) Format: free

(Include if Line A.1:INTEGRATION_TYPE = `GENERAL AREA`.*)*

Y_VERTICES = The latitude in decimal degrees of one vertex of the polygon for this scattering region.

X_VERTICES = The longitude in decimal degrees of one vertex of the polygon for this scattering region.

Note: Repeat this line for all vertices. Enter (-999, -999) to stop.

Note: The order of the vertices should be counterclockwise around the polygon.

Include lines I.5 – I.7 when modeling surface reverberation.

Line I.5: Format: A30

SCATTERING_MODEL = Flag indicating which surface scattering model to use.

`OGDEN-ERSKINE`	:= Ogden-Erskine
`CHAPMAN-HARRIS`	:= Chapman-Harris
`CONSTANT`	:= Constant scattering strength
`USER SUBROUTINE`	:= Use subroutine USER_SURF_SCAT.
A file name	:= Read in scattering table from a file.

```
Scattering Table
 0.0   -61.00
 5.0   -50.30
 9.5   -41.02
30.0   -19.10
60.0   -15.00
90.0   -10.00
```

At left is a sample scattering table file. The first line is a (maximum 80-character) title describing the contents of the file. Following this should come pairs of average grazing angle (°) and scattering strength (dB). There can be from 2 to 181 pairs. The angles are assumed to be in ascending order and range between 0° and 90°. Since BIREV will perform linear interpolation for missing angles, values for 0° and 90° should always be included.

Line I.6: Format: free

(Include if SCATTERING_MODEL = `OGDEN-ERSKINE` *or SCATTERING_MODEL =* `CHAPMAN-HARRIS`*.)*

WIND_SPEED = Wind speed, in knots.

Line I.7: Format: free

(Include if SCATTERING_MODEL = `CONSTANT`*.)*

CONSTANT_SS = The angularly independent surface scattering strength, in dB. Its sign should be negative.

Include lines I.8 – I.10 when modeling bottom reverberation.

Line I.8: Format: A30

SCATTERING_MODEL = Flag indicating which bottom scattering model to use.

`LAMBERTS LAW`	:= Lambert's Law .
`ROCK`	:= Urick's rock scattering model.
`SAND`	:= Urick's sand scattering model.
	:= More options on next page.

`SILT`	:=	Urick's silt scattering model.
`CLAY`	:=	Urick's clay scattering model.
`CONSTANT`	:=	Constant scattering strength.
`USER SUBROUTINE`	:=	Use subroutine USER_BOT_SCAT.
A file name	:=	Read in scattering table from a file. The format of this file is the same as for surface scattering strengths on Line I.5.

Line I.9: Format: free

(Include if SCATTERING_MODEL = `LAMBERTS LAW`*.)*

LAMBERTS_LAW_MU = The scattering strength coefficient at normal incidence, in dB. The value must be negative. Use −27 dB for MacKenzie scattering.

Line I.10: Format: free

(Include if SCATTERING_MODEL = `CONSTANT`*.)*

CONSTANT_SS = The angularly independent bottom scattering strength, in dB. Its sign should be negative.

Include lines I.11 – I.12 when modeling volume reverberation.

Line I.11: Format: free

SCATTERING_MODEL = Flag indicating which volume scattering model to use.

`CONSTANT`	:=	Constant scattering strength.
`USER SUBROUTINE`	:=	Use subroutine USER_VOL_SCAT.
A file name	:=	Read in scattering table from a file. The format of this file is the same as for surface scattering strengths on Line I.5.

Line I.12: Format: free

(Include if SCATTERING_MODEL = `CONSTANT`*.)*

CONSTANT_SS = The angularly independent volume scattering strength, in dB. Its sign should be negative.

Input Lines I.13 – I.14 if modeling a bistatic geometry and the chosen scattering model is not constant scattering strength.

Line I.13: Format: free

NU_IN_dB = The strength of the facet scattering for the Kirchhoff facet approximation for bistatic scattering, in dB. The value must be negative.

Line I.14: Format: free

SIGMA_IN_DEGREES = The angular spread factor for the Kirchhoff facet approximation for bistatic scattering, in degrees.

If CHOICE_OF_NUM_SCAT_PROCESSES = `MULTIPLE SCATTERING PROCESS` then process Lines I.5 – I.14

one more time for the "default" scattering model.

Include Lines J.1 – J.4 if
Line A.1:INTEGRATION_TYPE = `GENERAL AREA`.

Line J.1: Format: free

CHOICE_OF_INTEGRATION(SOURCE) = A flag denoting whether the user wants to specify the source's azimuthal integration limits or allow them to default to 360°.

- `DEFAULT INTEGRATION RANGE` := Default to 360°.
- `SPECIFY INTEGRATION RANGE` := User will specify the limits.

Line J.2: (2 Entries) Format: free

(Include if CHOICE_OF_INTEGRATION(SOURCE) = `SPECIFY INTEGRATION RANGE`.*)*

MIN_ANGULAR_INTEGRATION_ANGLE(SOURCE) = The minimum azimuthal bearing of integration about the source.

MAX_ANGULAR_INTEGRATION_ANGLE(SOURCE) = The maximum azimuthal bearing of integration about the source.

These bearings are centered on the source's location and specified in degrees clockwise from true north, ranging from 0° to 360°.

Line J.3: Format: free

CHOICE_OF_INTEGRATION(RECEIVER) = A flag denoting whether the user wants to specify the receiver's azimuthal integration limits or allow them to default to 360°.

- `DEFAULT INTEGRATION RANGE` := Default to 360°.
- `SPECIFY INTEGRATION RANGE` := User will specify the limits such as limiting to the port or starboard side of a horizontal receiver.

Line J.4: (2 Entries) Format: free

(Include if CHOICE_OF_INTEGRATION(RECEIVER) = `SPECIFY INTEGRATION RANGE`.*)*

MIN_ANGULAR_INTEGRATION_ANGLE(RECEIVER) = The minimum azimuthal bearing of integration about the receiver.

MAX_ANGULAR_INTEGRATION_ANGLE(RECEIVER) = The maximum azimuthal bearing of integration about the receiver.

These bearings are centered on the receiver's location and specified in degrees clockwise from true north, ranging from 0° to 360°.

LINE 12: (2 Entries) Format: free

FIRST_MULTIPATH(SOURCE) = The lower limit of source multipaths to be used for pairing with receiver multipaths. Must be greater than or equal to one.

LAST_MULTIPATH(SOURCE) = The upper limit of source multipaths to be used for pairing with receiver multipaths. Must be greater than or equal to the lower limit.

If greater than the number available, then all will be used.

Note that these have been sorted by TLGRID.

LINE 13: (2 Entries) Format: free

FIRST_MULTIPATH(RECEIVER) = The lower limit of receiver multipaths to be used for pairing with source multipaths. Must be greater than or equal to one.

LAST_MULTIPATH(RECEIVER) = The upper limit of receiver multipaths to be used for pairing with source multipaths. Must be greater than or equal to the lower limit. If greater than the number available, then all will be used.

Note that these have been sorted by TLGRID.

Line K: Format: free

(Include if Line 9:RVB_OR_TE_FLAG = `REVERBERATION` *or* `GENERAL TARGET ECHO` *and Line A.1:INTEGRATION_TYPE =* `GENERAL AREA`*.)*

EPSILON = The relative error specification that controls the integration optimization algorithm. Its value may range from 0 to 1. The closer to 0, the higher the accuracy of the prediction and the longer the execution time. If equal to 0, the algorithm is not used.

Line L: Format: A

(Include if source/receiver geometry is bistatic.)

TFNAME = Flag indicating whether to include the direct blast from the source to the receiver.

- `NO DIRECT BLAST` := Do not include direct blast.
- A file name := The name of a target file output from TLGRID. The target depth used must correspond to the receiver's depth, the location in this file must correspond to the source's location, and the radial bearing in the file must be within one-half degree of the bearing from the source to the receiver.

LINE 14: Format: free

GO = Flag indicating whether to continue directly into the integration phase of BIREV or to halt the program.

- `CONTINUE` := Continue program execution.
- `STOP` := Halt program execution after creating log and print files.

BIREV Output:

- A log file containing a listing of all the inputs provided to the program. This file can be used as input to the program.
- A print file PNTFIL summarizing and annotating the inputs provided to the program.
- Results written by subroutine WRITE_BIREV_OUTPUT. This routine outputs predictions to an output file (or files) in formats usable by further processing/plotting programs. The type of prediction requested dictates the number of output files and the format of those files.

If a "Single Bearing Calculation" was performed and the user chose not to decompose the reverberation by any angle, a single output file containing the reverberation time series is written out in the format read by program ACTENV. If the reverberation has been decomposed by some angle, then a single output file containing the reverberation is written out in a format read by program ANGPLOTS.

If a general integration was performed, the output format depends upon the angle by which the reverberation was distributed. If the reverberation was distributed by one array's beam pattern angles, the output file(s) is(are) written in a format read by program POSTBIREV. (Note: Multiple output files will be written if multiple beam patterns were specified for the *other* array. For example, if the reverberation was distributed by the receiver's beam pattern angles and the user specified *three* different source beam patterns, *three* output files will be written—one output file per source beam pattern.) If the reverberation is distributed by any other angle, a single output file is written in a format read by program ANGPLOTS.

For target echo predictions, the output format is that read by program POSTBIREV. Single target echo predictions will produce only one output file because, although the receiver's beam pattern was not applied during program execution, still only *one* source beam pattern was applied. General target echo predictions will produce two output files. One of these files contains peak target echo over the area of interest. The other output file contains average target echo over the area of interest. With both of these, the receiver's beam pattern will not have been applied.

The names of the output files are generated using the following rules:

- If there is only one output file, its name consists of the character string ".DAT" appended to the root name for reverberation output files (as input by the user).
- For general target echo predictions, one output file's name consists of the character string "_PEAK.DAT" appended to the root name; this output file contains the peak target echo over the area of interest. The other output file's name consists of the character string "_AVG.DAT" appended to the root name; this output file contains the average target echo over the area of interest.
- For reverberation predictions in which the user distributed the reverberation by one array's beam pattern angles, there is the possibility that multiple beam patterns were applied to the other array. If this is so, then one output file will be written for each of these beam patterns. Each output file's name consists of either
  - the integer representation of the main beam steering angle plus the character string ".DAT" appended to the root name, or
  - the beam number of the main beam plus the character string ".DAT" appended to the root name. For example, suppose the user distributed the reverberation by the source array's beam pattern angles and specified two receiver beam patterns. Suppose these beam pattern's main response lobes were designated by steering angles, e.g., 90° and 37.5°. The two output files would be named *rootname*_90.DAT and *rootname*_37.DAT, respectively. On the other hand, suppose these beam pattern's main response lobes were designated by beam numbers, e.g., 10 and 20. The two output files would be named *rootname*_10.DAT and *rootname*_20.DAT, respectively.

A8 Program TLVSR

TLVSR is a program to compute the transmission loss from the source (receiver) to a target/layer depth or boundary as a function of range from the source (receiver). The input files are those produced by TLGRID. Transmission loss vs range may be computed for a single bearing or for the entire TLGRID file. The single-bearing data files of transmission loss from the source-to-target and (by using reciprocity) from the target-to-receiver can be used in ACTENV to evaluate Eq.(9) for target returns as a function of range for monostatic and quasi-monostatic geometries. If the entire TLGRID file is processed and mapped into plan view, then a wide-area assessment of target returns can be calculated using the sonar equation representation (see Eq. (34) in Section 4.7 on page 34 of the report). Additionally, TLVSR will compute and print the ray arrival structure, as a function of range.

TLVSR Input Variables:

LINE 1: Format: A80

INFILE = Name of input data file created by TLGRID.
(Maximum of 80 characters.)

LINE 2: Format: A80

ANALYSIS_FLAG = Flag indicating type of calculation(s) to be performed by the program.

- `SINGLE BEARING` := Process single bearings for plotting transmission loss vs range, input to ACTENV for calculating target returns, or examining the arrival structure in each range bin.
- `ALL BEARINGS` := Process all bearings in the TLGRID file and produce transmission loss vs range and bearing plots in either a color waterfall or a plan view format.

Input Lines A.1 – A.4 if ANALYSIS_FLAG = `SINGLE BEARING`.

Line A.1: Format: free

RADIAL_BRG_TO_CALCULATE_TLVSR = Bearing to calculate transmission loss for.

Note: Bearing must equal value stored in file INFILE.

Line A.2: Format: A80

ACTENV_FLAG = Flag indicating that transmission loss vs range for input to ACTENV is to be calculated.

- `NO ACTENV` := Do not calculate results for use by ACTENV.
- A file name := File to store transmission loss vs range for input to ACTENV. *(Maximum of 80 characters.)*

Line A.3: Format: A80

PLOT_FLAG = Flag indicating that coherent and incoherent transmission loss vs range is to be calculated and plotted.

$\begin{cases} \texttt{NO PLOT} := \text{Do not plot.} \\ \text{A file name} := \text{File to store results for plotting. } \textit{(Maximum of 80 characters.)} \end{cases}$

Line A.4: Format: A80

PRTFIL = Flag indicating that the arrival structure for this bearing should be written to file. Note: This can be a large file.

$\begin{cases} \texttt{NO ARRIVAL STRUCTURE} := \text{Do not save the arrival structure to file.} \\ \text{A file name} := \text{File to store the arrival structure for each range bin. } \textit{(Maximum of 80 characters.)} \end{cases}$

Input Lines B.1 – B.5 if ANALYSIS_FLAG = `ALL BEARINGS`.

Line B.1: Format: A80

WATERFALL_FLAG = Flag indicating that transmission loss vs range and bearing should be plotted in a color waterfall format.

$\begin{cases} \texttt{NO PLOT} := \text{Do not plot.} \\ \text{A file name} := \text{File to store results for plotting. } \textit{(Maximum of 80 characters.)} \end{cases}$

Line B.2: Format: A80

PLANVIEW_FLAG = Flag indicating that transmission loss vs range and bearing should be plotted in a plan view format. The picture is created by setting up a rectangular grid made up of a user-specified number of pixels in the x and y directions. The bearing and range from the array of each pixel are calculated, and this information is used to look up the appropriate transmission loss values from the input data file.

$\begin{cases} \texttt{NO PLOT} := \text{Do not plot.} \\ \text{A file name} := \text{File to store results for plotting. } \textit{(Maximum of 80 characters.)} \end{cases}$

Note: For the remaining inputs, if several plots are to be made, then consistency is recommended. This will allow simple utility programs to manipulate the files at a later time.

Line B.3: (2 Entries) Format: free

(Include if PLANVIEW_FLAG = A file name.)

LATITUDE_MIN = Minimum latitude to view in plan view.

LATITUDE_MAX = Maximum latitude to view in plan view.

Line B.4: (2 Entries) Format: free

(Include if PLANVIEW_FLAG = A file name.)

LONGITUDE_MIN = Minimum longitude to view in plan view.

LONGITUDE_MAX = Maximum longitude to view in plan view.

Line B.5: Format: free

(Include if PLANVIEW_FLAG = A file name.)

PIXELS_PER_KM = The number of pixels per kilometer to be used in constructing the plan view plot.

TLVSR Output:

For a single bearing, the possible output from program TLVSR consists of:

- a data file (ACTENV_FLAG) of incoherent transmission loss as a function of range for input to ACTENV,
- a data file (PLOT_FLAG) of incoherent and coherent transmission loss as a function of range for plotting.
- a print file (PRTFIL) of ray arrival structure as a function of range.

If the entire TLGRID file is processed (i.e., all bearings), the possible output from program TLVSR consists of:

- a data file (WATERFALL_FLAG) of incoherent transmission loss as a function of range and bearing suitable for plotting in a waterfall format,
- a data file (PLANVIEW_FLAG) of incoherent transmission loss as a function of location (i.e., range and bearing have been mapped into a plan view). The wide-area assessment of target returns can be constructed from these files using utility programs to combine and scale the results.

A9 Program ACTENV

ACTENV is a program that summarizes the performance envelope of the acoustic systems operating characteristics. Expressing beam power as a function of range or time, ACTENV will scale and plot the surface and bottom reverberation, ocean ambient noise, and target echo envelopes. Different reverberation envelopes can be combined in a weighted average to produce a composite level. Alternatively, envelopes can be time-averaged to more accurately represent system characteristics. Source and receiver array properties, boundary spectral spreading, and processing parameters are some of the more important input requirements.

ACTENV Inputs:

LINE 1: Format: 18A4

TITL(I), (I=1,17) = Title (up to 68 characters) to be used by plotting routines. (Formatted according to DISSPLA requirements.)

LINE 2: Format: A30

PRTFIL = Name of output print file.
(Maximum of 30 characters.)

LINE 3: Format: free

IPING = Ping type

$$\begin{cases} 1 := \text{Gated-CW} \\ 2 := \text{Impulsive} \\ 3 := \text{FM - See LINE 8:SIGNALS\_DURATION in program BIREV.} \end{cases}$$

Line A: Format: free

(Include when IPING = 1 or IPING = 3 .)

DUR = Ping duration (s).

Line B: Format: free

(Include when IPING = 2 or IPING = 3 .)

BW = Bandwidth (Hz).
For impulsive (IPING=2) this is the analysis bandwidth.
For FM (IPING=3) this is the pulse's bandwidth, assumed to be the analysis bandwidth.

LINE 4: (2 Entries) Format: free

TMIN = Minimum time (seconds after transmission) for which envelopes will be calculated.
Default = 0.

TMAX = Maximum time (s) for which envelopes will be calculated.
Default = 100000.

LINE 5: (4 Entries) Format: free

SLE = Source level per element (dB).
– must be input.

ELN = Number of source elements. Should correspond with number of elements used to create the source beam pattern in program TLGRID or BIREV. The total source level is calculated in dB using $SL = SLE + 20\ \log_{10}(ELN)$.

ANL = Omnidirectional ambient noise level (dB/Hz).

RDI = Horizontal receiver directivity index (dB). Should be zero for omnidirectional and vertical array receivers. For fully-populated, unshaded linear arrays with N elements, $RDI \approx 10 * \log_{10}(N)$.

LINE 6: (4 Entries) Format: free

TAXIS = Length (in) of time axis of plot (≤ 10).
Default = (TMX−TMN)×0.05

TMN = Start time (s) for plot of envelope.
Default = TMIN (see LINE 4).

DT = Time increments (s/in) of plot labels.
Default = automatic scaling.

TMX = End time (s) for plot of envelope.
Default = TMAX (see LINE 4).

LINE 7: (4 Entries) Format: free

ZAXIS = Length (in) of envelope axis of plot. (≤ 8)
Default = computed by program.

ZMIN = Minimum power (dB) plotted.
Default = computed by program.

DZ = dB labels per inch on plot.

Default = automatic scaling.
ZMAX = Maximum power (dB) plotted.
Default = computed by program.

LINE 8: (2 Entries) Format: free
NOREV = Number of surface reverberation envelopes to be read in.
IFWGT = Flag indicating that envelopes will have nonuniform weightings.
Default = 0.

For Surface Reverberation, repeat Lines C.1–C.2 NOREV Times.

Line C.1: Format: A30
(Include when NOREV > 0 for surface reverberation files.)
REVFIL = Name of input surface reverberation data file.
(Maximum of 30 characters.)

Line C.2: Format: free
(Include when IFWGT > 0.)
WGTT = Weight for this surface reverberation envelope.

LINE 9: (2 Entries) Format: free
NOREV = Number of bottom reverberation envelopes to be read in.
IFWGT = Flag indicating that envelopes will have nonuniform weightings.
Default = 0.

For Bottom Reverberation, repeat Lines D.1–D.2 NOREV Times.

Line D.1: Format: A30
(Include when NOREV > 0 for bottom reverberation files.)
REVFIL = Name of input bottom reverberation data file.
(Maximum of 30 characters.)

Line D.2: Format: free
(Include when IFWGT > 0 .)
WGTT = Weight for this bottom reverberation envelope.

LINE 10: Format: free
IFECHO = Flag indicating that target echo levels are to be computed.
Default = 0.

Include Lines E.1–E.6 if IFECHO > 0 .

Line E.1: Format: A30
TRLFIL = File name for transmission loss from target to receiver.

(Maximum of 30 characters.)

Line E.2: Format: free

LSTAPE = Flag indicating that additional file must be read in for transmission loss from source to target.

Default = 0.

Note: Needed for quasi-monostatic or bistatic configuration.

Line E.3: Format: A30

(Include when LSTAPE > 0 .)

TSLFIL = File name for transmission loss from source to target.

(Maximum of 30 characters.)

Line E.4: Format: free

SEP = Horizontal range (km) from source to receiver.

Default = 0.

Line E.5: Format: free

BRG = Target bearing (deg) relative to receiver.

Default = 0.

Line E.6: Format: free

TGTS = Target strength (dB/m^2).

ACTENV Output:

- Printout on PRTFIL containing: signal and system parameters; weighting summary; surface and bottom boundary reverberant power vs time; and target echo level vs time.
- Plots:
  1. Bottom and surface reverberation envelopes as function of time or range.
  2. Target echo levels as function of time or range.

A10 Program POSTBIREV

POSTBIREV is a program to apply the beam patterns to the reverberation and target-echo angle-time series produced by BIREV for a multibeam receiver. POSTBIREV is also used to perform various manipulations of the files containing the time series, such as calculating the total volume reverberation by summing the contributions from each layer, applying a time averaging window, mimicking a ping repetition rate, shifting the origin of the time axis, and constructing multistatic results. It can also be used to extract single beams or bearings for input to ACTENV. The program REVPLOTS is used to scale the output time series from POSTBIREV to absolute levels and plot them in a waterfall or plan-view format.

The program is a set of subroutines driven by a menu. This enables the user to string a sequence of operations together. Most of the subroutines require only a few inputs that will be obvious to the user, i.e., file names and one or two numbers. Thus, these inputs will not be described here. The exception is the subroutine to calculate the beam pattern tables, but this subroutine usually is executed only once. The input files to POSTBIREV are the raw (uncalibrated) times series calculated by BIREV or its own output files.

Subroutine CREATE_BP_DATABASE_FILE: Creates a lookup table of beam patterns for a specific array. The number of beam patterns can be based on steering angles or beams equally spaced in "cosine space." Each beam pattern is sampled at 0.5°. This resolution is generally sufficient for the 128-element arrays typically employed. The majority of inputs for this subroutine are described in the inputs to program BIREV, Section A7 on Lines G.1 – G.18.

Subroutine CREATE_BP_PRINT_FILE: Similar to the previous subroutine except it calculates a specific beam pattern and writes it to a print file.

Subroutine BEAMFORM_RAW_BIREV_DATAFILE: Apply all the beam patterns contained in a beam pattern database file to the reverberation data contained in a raw BIREV datafile. Sum the reverberation vs time and launch/arrival angle data over launch/arrival angles for each time increment, thus reducing the data to reverberation vs time vs beam (or steering angle).

Subroutine SUM_RAW_BIREV_DATAFILES: Take two or more raw BIREV datafiles, possibly of different lengths with respect to time series limits, add them together, and output their sum in raw BIREV datafile format.

Subroutine SUM_BEAMFORMED_BIREV_DATAFILES: Take two or more beamformed BIREV datafiles, possibly of different lengths with respect to time series limits, add them together, and output their sum in beamformed BIREV datafile format.

Subroutine COLLAPSE_RAW_BIREV_FILE: When no beam pattern is applied in BIREV, its output has generally been distributed over an angle measured with respect to the source or receiver array. This subroutine sums the raw BIREV data over its angular distribution and outputs a single time series, as though the source or receiver array was omnidirectional.

Subroutine RESTRICT_ANALYSIS_TIME_WINDOW: Extract a portion of the time series from each beam in a beamformed BIREV datafile.

Subroutine ADD_IN_NOISE_LEVEL: Adds a constant noise level to the modeled reverberation contained within a beamformed BIREV datafile.

Subroutine APPLY_TIME_AVERAGING_WINDOW: Takes a beamformed BIREV datafile and applies a forward-looking, user-specified time-averaging window (s) to the modeled reverberation. Output the results in beamformed BIREV datafile format.

Subroutine MIMIC_PING_REPETITION_RATE: Takes a beamformed BIREV datafile and mimics ping repetition rate by adding a portion of the file onto itself via a user-input ping repetition rate (s).

Subroutine SCALE_AND_SHIFT_BF_DATA: Allows the user to scale the modeled reverberation in a beamformed BIREV datafile by a specified number of dB and shift it in time by a specified number of seconds. After scaling and shifting, writes out the result in beamformed BIREV datafile format.

Subroutine EXTRACT_BEAMFORMED_TIME_SERIES: Extracts time series from a beamformed BIREV datafile for user-specified beams and writes them to an ASCII print file that can be read by program ACTENV and plotted.

A11 Program REVPLOTS

REVPLOTS processes the reverberation or target echo time series contained in files created by BIREV or POSTBIREV and outputs color waterfall plots of received level vs beam/bearing and time or plan view plots of received level vs apparent range and azimuth with respect to the receiver.

REVPLOTS Input Variables:

LINE 1: Format: A80

ANALYSIS_FLAG = Flag indicating type of calculation(s) to be performed by the program. Resulting output are files for display using a color-graphics plotting package.

- `WATERFALL` := Display received level vs beam/bearing and time.
- `AMBIGUOUS PLAN VIEW` := Display received level in plan view at apparent range and azimuth with respect to the receiver. Only one input file will be requested and its contents will be mirror-imaged about the heading of the receiver.
- `RESOLVED PLAN VIEW` := Display received level in plan view at apparent range and azimuth with respect to the receiver. Two input files will be requested, one for the port (left) side of the receiver and one for the starboard (right) side.

LINE 2: Format: A80

INFILE1 = Name of input data file created by BIREV or POSTBIREV. If ANALYSIS_FLAG = `RESOLVED PLAN VIEW` then this file should contain the received level for the port side of the receiver. *(Maximum of 80 characters.)*

LINE 3: Format: free

IPING = Ping type

- 1 := Gated-CW
- 2 := Impulsive
- 3 := FM

Line A: Format: free

(Include when IPING = 1 or IPING = 3 .)

DUR = Ping duration (s).

Line B: Format: free

(Include when IPING = 2 or IPING = 3 .)

BW = Bandwidth (Hz).
For impulsive (IPING=2) this is the analysis bandwidth.
For FM (IPING=3) this is the pulse's bandwidth, assumed to be the analysis bandwidth.

LINE 4: (2 Entries) Format: free

SLE = Source level per element (dB).

ELN = Number of source elements. Should correspond with number of elements used to create the source beam pattern in program TLGRID or BIREV. The total source level is calculated in dB using $SL = SLE + 20\ \log_{10}(ELN)$.

Line C: Format: A80

(Include if **ANALYSIS_FLAG** = `RESOLVED PLAN VIEW`.*)*

INFILE2 = Name of input data file created by BIREV or POSTBIREV. This file should contain the received level for the starboard side of the receiver. *(Maximum of 80 characters.)*

Input Lines D.1-D.7 if ANALYSIS_FLAG = `AMBIGUOUS PLAN VIEW` or `RESOLVED PLAN VIEW`.

Line D.1: (2 Entries) Format: free

SOURCE_LATITUDE = The source arrays's latitude, in decimal degrees, and should range between -90° and 90°, south and north, respectively.

SOURCE_LONGITUDE = The source array's longitude, in decimal degrees, and should range between -180° and 180°, east and west, respectively.

Line D.2: (3 Entries) Format: free

RECEIVER_LATITUDE = The receiver arrays's latitude, in decimal degrees, and should range between -90° and 90°, south and north, respectively.

RECEIVER_LONGITUDE = The receiver array's longitude, in decimal degrees, and should range between -180° and 180°, east and west, respectively.

RECEIVER_HEADING = The heading of the receiver array. This bearing is specified in degrees clockwise from true north, ranging from 0° to 360°.

Line D.3: (2 Entries) Format: free

LATITUDE_MIN = Minimum latitude in decimal degrees to view in plan view.

LATITUDE_MAX = Maximum latitude in decimal degrees to view in plan view.

Line D.4: (2 Entries) Format: free

LONGITUDE_MIN = Minimum longitude in decimal degrees to view in plan view.

LONGITUDE_MAX = Maximum longitude in decimal degrees to view in plan view.

Line D.5: Format: free

SPEED_OF_SOUND = Speed of sound (km/s) used to convert time into range.

Line D.6: Format: free

ADDITIONAL_TIME_SHIFT = Additional adjustment to time series (s) in file to account for the source/receiver separation. Program will attempt to calculate time of first arrival from source to receiver and use it to create plot. This input allows the user to make fine adjustments if necessary.

Line D.7: Format: free

PIXELS_PER_KM = The number of pixels per kilometer to be used in constructing the plan view plot.

REVPLOTS Output:

- A color waterfall plot of reverberation vs beam/bearing and time.
- A color plan view plot of reverberation vs angle and apparent range from the receiver.

A12 Program ANGPLOTS

This program reads in a file produced by BIREV that contains an angular distribution of returns. This file may contain reverberation or single target echo. Scale factors similar to those discussed for program ACTENV in Section A9 on page 83 are applied. The output file consists of reverberation (target echo) vs the angle selected during execution of BIREV (e.g., grazing angle) and time.

ANGPLOTS Inputs:

LINE 1: Format: A80

ANGFIL = Name of input file from BIREV

(Maximum of 80 characters.)

LINE 2: Format: A80

PRTFIL = Name of output file for plotting.

(Maximum of 80 characters.)

LINE 3: Format: free

IPING = Ping type

$$\begin{cases} 1 := \text{Gated-CW} \\ 2 := \text{Impulsive} \\ 3 := \text{FM} \end{cases}$$

Line A: Format: free

(Include when IPING = 1 or IPING = 3 .)

DUR = Ping duration (s).

Line B: Format: free

(Include when IPING = 2 or IPING = 3 .)

BW = Bandwidth (Hz).

For impulsive (IPING=2) this is the analysis bandwidth.
For FM (IPING=3) this is the pulse's bandwidth, assumed to be the analysis bandwidth.

LINE 4: (3 Entries) Format: free

SLE = Source level per element (dB).

ELN = Number of source elements. Should correspond with number of elements used to create the source beam pattern in program TLGRID or BIREV. The total source level is calculated in dB using $SL = SLE + 20\ \log_{10}(ELN)$.

RDI = Horizontal receiver directivity index (dB). Should be zero for omnidirectional and vertical array receivers or if the full receiver beam pattern was applied in BIREV. For fully-populated, unshaded linear arrays with N elements, $RDI \approx 10 * \log_{10}(N)$.

ANGPLOTS Output:

The output from program ANGPLOTS consists of a single data file of reverberation (or single target echo) as a function of some angle (i.e., grazing angle) and time in a format suitable for plotting in a waterfall format.